DEVELOPMENT OF AN EFFICIENT VISCOUS APPROACH IN A CARTESIAN GRID FRAMEWORK AND APPLICATION TO ROTOR-FUSELAGE INTERACTION

A Thesis
Presented to
The Academic Faculty

by

Jae-doo Lee

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in Aerospace Engineering

Georgia Institute of Technology
August 2006

DEVELOPMENT OF AN EFFICIENT VISCOUS APPROACH IN A CARTESIAN GRID FRAMEWORK AND APPLICATION TO ROTOR-FUSELAGE INTERACTION

Approved by:

Dr. Stephen M. Ruffin, Advisor
School of Aerospace Engineering
Georgia Institute of Technology

Dr. Suresh Menon
School of Aerospace Engineering
Georgia Institute of Technology

Dr. Lakshmi N. Sankar
School of Aerospace Engineering
Georgia Institute of Technology

Dr. Marc K. Smith
School of Mechanical Engineering
Georgia Institute of Technology

Dr. Marilyn J. Smith
School of Aerospace Engineering
Georgia Institute of Technology

Date Approved: May 4, 2006

Dedicated to my Grandfather
who had been very proud of me
to be a Ph.D. in Aerospace Engineering
until he passed away in 2003

ACKNOWLEGEMENTS

It already has been five years since I first came to Georgia Tech to follow my dream. It was not easy to adapt myself to a strange circumstance, but I could complete my graduate career thanks to everyone's help around me. First of all, I wish to express a great appreciation to my advisor, Dr. Stephen Ruffin, who have guided and supported me for five years. I still remember his smile at the moment I fist met him. Since then, he has showed me unchanging trust and full of consideration for me, which always encourage me to complete my thesis. I would also like to thank my thesis committee members, Dr. Lakshmi Sankar, Dr. Suresh Menon, Dr. Marilyn Smith, and Dr. Marc Smith for providing valuable suggestions on my research.

I want to say thank you to Jinwook Lee for sharing all the moment at Georgia Tech and many discussions. He is like my brother and has been together with me for five years, living at the same apartment and studying at the same Lab. It would be unforgettable in my whole life. All the other former and current colleagues from ARTLab (Aerothermodynamics Research and Technology Laboratory) at Georgia Tech gave me the most colorful memories as well as helps for my thesis. To Dr. David Marshall, Mr. Susheel Kumar Sekhar, Mr. Tim Eymann, Mr. Srujal Patel, and Mr. Joel MaloMolina, I like to express my appreciation. I would also like to thank my fellow graduate student, Mr. David O'Brien for providing reference data and papers from his preceding research.

Finally, I want to express my gratitude with all my heart to my parents for providing all the support and devotion throughout my whole life. Most of all, I cannot thank enough to my wife, Eunjung, for all the love and encouragement during my graduate years.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS

Alphanumeric

A_0	Blade collective pitch angle
A_1	Blade lateral cyclic pitch angle
a	Speed of sound
B	Law-of-the-wall constant
B_1	Blade longitudinal cyclic pitch angle
C_d	Sectional drag coefficient
C_f	Skin friction coefficient
C_l	Sectional lift coefficient
C_p	Specific heat at constant pressure, pressure coefficient
C_T	Thrust coefficient
C_1, C_2	Constants of Sutherland's law
$C_\mu, C_{\varepsilon 1}, C_{\varepsilon 2}$	Constants of k-ε turbulence model
D	Drag
e	Internal energy per unit mass
E_t	Total energy per unit volume
$\boldsymbol{F}, \boldsymbol{G}, \boldsymbol{H}$	Flux vectors in x-, y-, and z-directions
I_t	Turbulent intensity
K_k, K_ε	Limiter constants of turbulent kinetic energy and dissipation rate
k	Turbulent kinetic energy
L	Characteristic length, Laplacian, Lift
M	Mach number
N	Number of rotor blade
$\boldsymbol{n}$	Unit vector
p	Static pressure

P	Production of turbulent kinetic energy
Pr	Molecular Prandtl number
Pr_t	Turbulent Prandtl number
q	Heat flux
R	Gas constant, Riemann invariant, Radius of rotor
$\boldsymbol{R}$	Residual vector
r	Recovery factor, radial position
Re	Reynolds number
$\boldsymbol{S}$	Vector of source terms
s	Fluctuating strain rate
T	Temperature, rotor thrust
t	Time
$\boldsymbol{U}$	State vector of conserved variables
V	Velocity magnitude, cell volume
V_F	Froude velocity
$\boldsymbol{W}$	Vector of primitive variables
u, v, w	Mean velocities in the x-, y-, and z-directions
u', v', w'	Fluctuating velocities in the x-, y-, and z-directions
u_τ	Friction velocity

Greek

α_e	Effective angle of attack
α_i	Induced angle of attack
α_t	Twist angle at the blade tip
β	Flap angle of rotor blade
δ_{ij}	Kronecker delta function
γ	Ratio of specific heat
ε	Rate of dissipation of turbulence energy or coefficient of MUSCL

	scheme
κ	Thermal conductivity, von Karman constant, or coefficient of MUSCL scheme
λ	Eigenvalue
μ	Rotor advance ratio
μ_l	Molecular viscosity
μ_t	Eddy viscosity
ν	Kinematic viscosity
Θ	Pitch angle of blade
ρ	Density
$\sigma_k, \sigma_\varepsilon$	Constants of Launder and Spalding's turbulence model
τ	Shear stress
Ω	Angular velocity
ψ	Blade azimuth angle

Superscript

$'$	Fluctuating turbulent properties
*	Nondimensionalized quantity
~	Roe's averaged value
n	N^{th} time step

Subscript

g	Ghost cell
i, j, k	Tensor notation or numbering of numerical stencil
l	Laminar or molecular properties
L	Properties based on characteristic length, Left hand side
max	Maximum item
N	Normal direction

R	Right hand side
ref	Reference point
T	Tangential direction
t	Turbulent properties
x , y , z	Components in x-, y-, and z-directions
w	Properties on solid wall
∞	Infinity, freestream values

SUMMARY

Despite the high cost of memory and CPU time required to resolve the boundary layer, a viscous unstructured grid solver has many advantages over a structured grid solver such as the convenience in automated grid generation and shock or vortex capturing by solution adaption. Since the geometry and flow phenomenon of a helicopter are very complex, unstructured grid-based methods are well-suited to model properly the rotor-fuselage interaction than the structured grid solver. In present study, an unstructured Cartesian grid solver is developed on the basis of the existing solver, NASCART-GT. Instead of cut-cell approach, immersed boundary approach is applied with ghost cell boundary condition, which increases the accuracy and minimizes unphysical fluctuations of the flow properties. The standard k-epsilon model by Launder and Spalding is employed for the turbulence modeling, and a new wall function approach is devised for the unstructured Cartesian grid solver. It is quite challenging and has never done before to apply wall function approach to immersed Cartesian grid. The difficulty lies in the inability to acquire smooth variation of y^+ in the desired range due to the non-body-fitted cells near the solid wall. Applying the conventional wall function wall boundary condition on the immersed Cartesian cells near the wall would result in a large magnitude of non-physical fluctuations of the flow properties, thereby, cause instability of the computation. The wall function boundary condition developed in this work yields stable and reasonable solution within the accuracy of the turbulence model. The grid efficiency is also improved with respect to the conventional method. The turbulence

modeling is validated and the efficiency of the developed boundary condition is tested in 2-D flow field around a flat plate, NACA0012 airfoil, axisymmetric hemispheroid, and rotorcraft applications.

For rotor modeling, an actuator disk model is chosen, since it is efficient and is widely verified in the study of the rotor-fuselage interaction. This model considers the rotor as an infinitely thin disk, which carries pressure jump across the disk and allows flow to pass through it. The full three dimensional calculations of Euler and RANS equations are performed for the GT rotor model and ROBIN configuration to test implemented actuator disk model along with the developed turbulence modeling. Finally, the characteristics of the rotor-fuselage interaction are investigated by comparing the numerical solutions with the experiments.

C H A P T E R I

INTRODUCTION

The overall objective in current research is to develop an efficient Navier-Stoke methodology that can model complex rotor-fuselage interactions. To do this, an unstructured Cartesian grid topology is chosen due to the convenience of automated grid generation and accuracy in vortex capturing. The new boundary approach is applied for laminar and turbulent capability, and the actuator disk modeling is employed for rotor analysis into the existing viscous Cartesian grid solver, called NASCART-GT. This chapter explains the fundamentals and backgrounds of the rotor-fuselage interaction and Cartesian grid framework, followed by the research objectives.

1.1 ROTOR-FUSELAGE INTERACTION

The interaction between the rotor and the helicopter fuselage generates complex flow field affecting vibration, acoustics and overall vehicle performance. If the performance of an isolated rotor and fuselage is analyzed separately, the interactional aerodynamics around the rotor-fuselage combination cannot be obtained by a simple linear superposition of each individual result because of the inherent nonlinear behavior of the flow. Therefore, the

accurate prediction of rotor-fuselage interaction is essential for optimum design and analysis of rotorcraft.

To study the mutual effect of rotor and fuselage, the flow around a rotor must be modeled properly. Over the years, many rotor models have been developed to analyze rotor-fuselage interaction based on Euler or Navier-Stokes solver. Whitfield and Jameson [96] studied the propeller-wing interaction by introducing a source term in the Euler equation. Rajagopalan and Mathur [69] modeled the rotor as an actuator disk to solve incompressible Navier-Stoke equations around the rotor. The actuator disk is an infinitely thin disk, which carries discontinuities of flow properties using the source terms in the momentum and energy equations or enforcing the pressure jump on the disk boundary. The previous authors applied the source type actuator disk model, and Zori and Rajagopalan [103] employed this method to simulate the rotor-fuselage interaction. The boundary condition type actuator disk method has also been successfully used for many structured and unstructured solvers [7, 12, 24, 66] for the rotor-fuselage interaction. Lately, O'Brien and Smith have published a series of papers [63-65, 71] in which they discuss various computational techniques on the rotor-fuselage interaction including the actuator disk methods of boundary type and source type in an unstructured grid RANS solver. Hariharan and Sankar [36] introduced an overset grid method to solve the rotor-fuselage interaction phenomenon. They used the stationary frame around the fuselage and rotating frame near the rotor to capture the unsteady flow motion. Though the overset grid method gives detailed unsteady analysis, it is computationally expensive.

The CPU time and computer memory resources required for full three-dimensional, unsteady, viscous calculation of the rotating blade with the fuselage using unstructured grid solver are extreme. Use of such an approach for multiple configurations considered in design or for a vehicle in maneuver is not practical. Meanwhile, the analysis using the actuator disk model, which has been widely verified in the analysis of the rotor-fuselage interaction is computationally less intensive than the full unsteady calculation, and may be suitable for some fuselage design and maneuver analysis. Two types of actuator disk formulations are common: boundary type and source type. Both approaches have been used very successfully for actuator disk modeling. In the present study, the boundary type actuator disk method is chosen, although both approaches are applicable to an unstructured Cartesian grid solver. Applying the boundary type actuator disk method in a Cartesian grid solver requires additional work in the grid generation. This is because the disk boundary should be aligned with the Cartesian cell faces, while the rotor location is independent of the grid in the source type method. However, the source type method is somewhat less robust and may give non-physical solution when the source spacing is coarser than the local grid spacing [63].

1.2 CARTESIAN GRID FRAMEWORKS

1.2.1 UNSTRUCTURED AND CARTESIAN GRIDS

The first step in a CFD process is to choose an appropriate grid topology, which can be categorized into either structured or unstructured types depending on the data structure representing the grid points. In general, the fuselage configuration of a helicopter is complex

and has many attachments. The flow field is also very complicated including vortex induced by the rotor and separation caused by bluff fuselage shape. Unstructured grid methods can easily employ automated grid generation for complex fuselage and solution adaption for vortex capturing. In spite of these advantages of the unstructured grid topology over the structured grid, until recently it could not be used for a viscous flow calculation due to the high cost of memory and CPU time inside the boundary layer. As high performance parallel computer systems have become more recently available, the unstructured grid solvers have been very popular even in the calculation of viscous and turbulent flow.

Many unstructured grid solvers use pyramid or prism type grid topology, in which the cells are body-fitted and it is easy to apply conservative integration for finite volume method. However, the numerical solution of the equations of fluid dynamics is simplified and the truncation error would be greatly reduced, if the discretization is performed in a Cartesian coordinate system. Many flow simulations involve complex geometries with curved and planar boundaries oblique to the grid. In the simplest Cartesian grid formulation, such boundaries are generally approximated as a series of staircase steps. Unless a very fine grid is used, the predicted velocity and pressure fields adjacent to the boundary will be inaccurate. To solve the problems related to the boundary treatment, many approaches have been developed in Cartesian grid solvers.

For an accurate simulation, the computational domain should include arbitrary cut cell near the solid body as shown in Figure 1. A common problem with cut cell is the creation of very small cells. This leads to problems with stiffness of the equations and non-physical fluctuations of flow variables near the body [55]. In case of time-dependent simulations, it

limits the time step and influences the stability. Researchers have dealt with this in a number of ways, including hybrid grid topology. The hybrid grid topology [14] employs the body-fitted structured grid near wall combined with Cartesian grid away from body as shown in Figure 2. Using the structured grid near wall, however, the laborious grid generation and complicated coordinate transformation are still required.

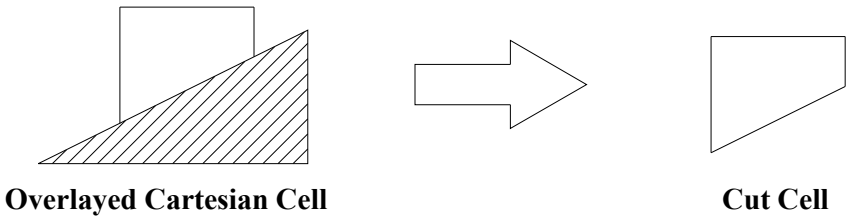


Figure 1 : Generated cut cell near solid body

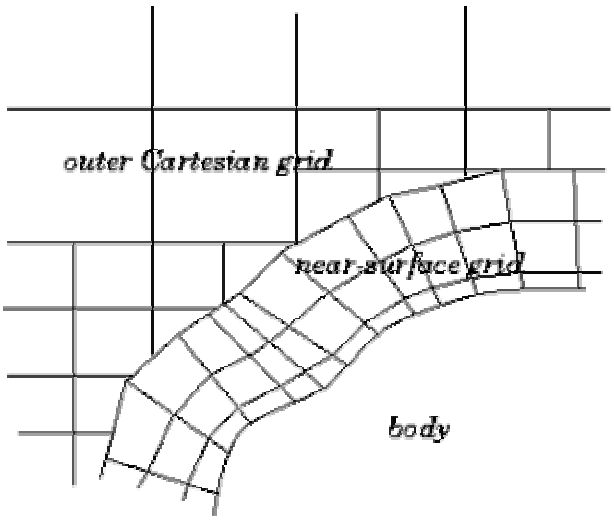


Figure 2 : Example of Cartesian hybrid grid topology from Ref. [14]

Alternative approaches use a Cartesian grid for all cells except those which are intersected by the boundary. This is called merged cut cell method shown in Figure 3. The merged cut cell method was presented by Udaykumar et al.[85, 86] and Ye et al.[98]. They truncated and merged the boundary cell into a nearest flow cell so that they conform to the shape of the boundary surface.

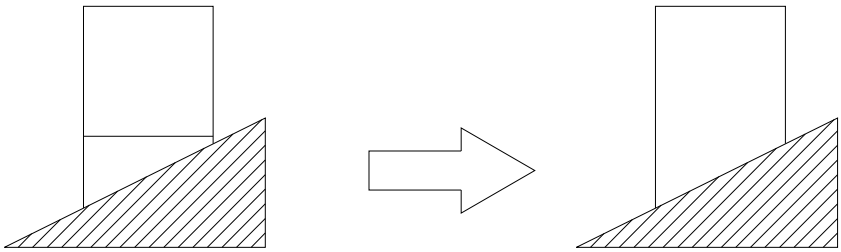


Figure 3 : Illustration of merged cell

Another method, called embedded boundary method, is to remove the surface cells from the finite volume formulation and extrapolate flow properties on the boundary cell center from the reference point. In this way, the small cut-cell problem can be solved and the advantages of a Cartesian grid are retained for the standard, non-boundary cells. Cells completely outside the flow domain are removed from the computations. The embedded boundary method was implemented to Cartesian grid solver, called NASCART-GT that is former version of current code, by Marshall and Ruffin [53, 54] in the computations of Euler and Navier-Stokes equations. Berger [6] introduced another method that employs a sub-cell resolution procedure for embedded Cartesian cells in his Euler solver. In this approach, the

flow properties at the cut-cell center are updated using different numerical stencil from the ordinary flow cells. His approach has been applied to a parallel computation and a moving grid technique for Euler calculation [3, 61]. However, it is still required in an embedded boundary cell method to relocate the cell center to the centroid of the truncated cell as shown in Figure 4.

The immersed boundary method using ghost cell is first introduced by Forrer and Jeltsch [27] for Cartesian grid in 1998. The boundary cell is not truncated and maintains cubic shape. As shown in Figure 4, the boundary cell center is not shifted to the cut-cell center and maintains at the centroid of non-cut cell independent of boundary shape. A. Dadone [17-21] successfully solved the 2D and 3D Euler equation for an unstructured Cartesian grid using the ghost-cell immersed boundary method. He considered solid walls as boundaries immersed in the flow field and enforces boundary conditions at ghost cell centers located inside the body in a position close to the wall. Symmetry conditions are enforced on the ghost cells, and the flow computations are performed using these ghost cell centers without taking into account the presence of the wall. It solved the problem of the misalignment of cell centers. It also eliminated the requirement of finding cell centroids thereby saving computer memory.

The merged cut cell method has an advantage over the immersed body method. Since the flow properties at the cut cell center are integrated from the wall boundary and flow cells, the state vector is always conservative. However, it generally entails a considerable increase in complexity, since fluxes between diagonally adjacent cells must also be calculated, and the computational molecule for merged boundary cells become different to that used for the

standard cells. Since the cut cell center is not aligned with other flow cells, the order of accuracy in spatial discretization is limited during integration of flow cells contacting the cut cell. This is also true for embedded boundary cell approach. From the reasons enlisted above, the immersed boundary approach is employed in the developed code

.

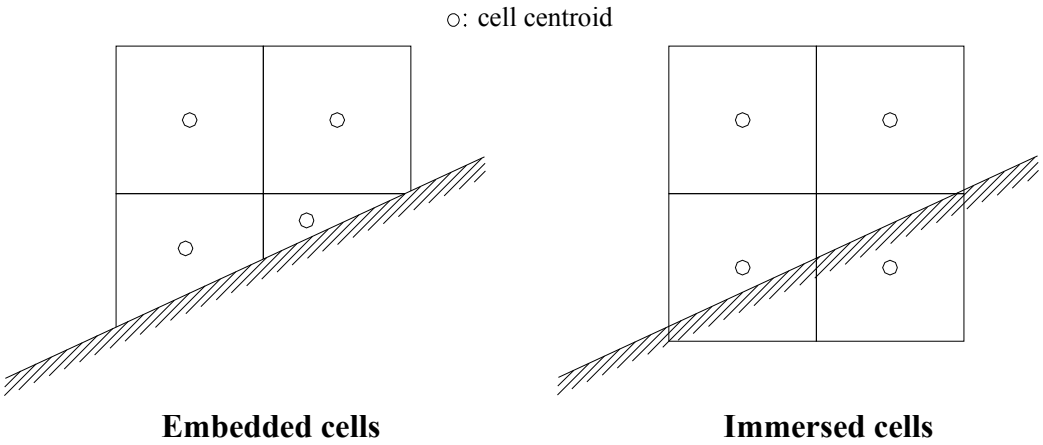


Figure 4 : Comparison of embedded and immersed cells

1.2.2 VISCOUS CALCULATION

Most unstructured Cartesian grid techniques have been developed to solve the Euler equations. Excluding the hybrid grid solver, there has been relatively little research for the full Navier-Stokes equation. Udaykumar et al.[85, 86] published papers in 1997 and 2001 in which they discuss merged cell approach in uniform, non-staggered grid to calculate two-dimensional Navier-Stokes equation in the presence of fixed/moving boundaries. Ye et

al.[98] applied a similar formulation but included an improved interpolation scheme at the boundaries.

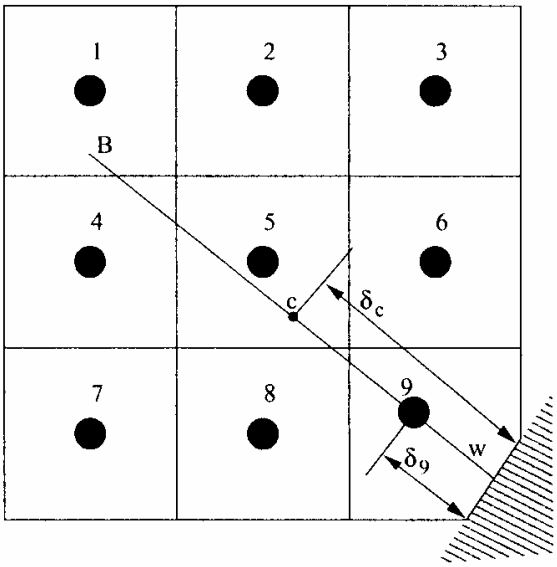


Figure 5 : Wall boundary treatment of embedded Cartesian grid from Ref. [54]

In 2002, Marshall [55] introduced the reference point method in unstructured, embedded Cartesian grids to solve the compressible Navier-Stokes equation, and successfully tested for a series of two-dimensional geometries. He used the linear extrapolation to find the primitive variables at the boundary cell center, which is relocated to the truncated cut-cell center as shown in Figure 5. To enforce the boundary condition at the cell center "9", the reference point "c" is specified. It is located on the ray normal to the wall for the given distance, δ_c, and its primitive variables are interpolated using 1st order distance weighted averaging from its neighbor cells. For the pressure and adiabatic wall temperature, the Neumann type

boundary conditions are satisfied by simply setting the pressure and temperature at point "9" are identical to those at the reference point if a locally flat wall is assumed. The constant wall temperature and no-slip condition, are enforced using the ratio of δ_c and δ_9 from the calculated primitive variables at the reference point.

Three-dimensional Navier-Stokes calculation is performed by Kirkpatrick [45] in non-uniform Cartesian grid. He employed the embedded Cartesian cells at the boundary surface, and created new cells that conform to the shape of the surface. In his approach, "cell-linking" method was developed to overcome problems associated with the small cells. However, it still involves the complexity in creation and linking of boundary cells. In 2003, Tseng and Ferziger [83] introduced host-cell immersed boundary method for calculating Navier-Stokes equation and simulating turbulent flows with LES solver in a uniform Cartesian grid.

While the methods mentioned above are based on the finite volume integration in an unstructured Cartesian grid, finite difference methods have been introduced by many researchers in a structured Cartesian grid solver. Peskin et al. [48, 57, 67, 72, 101] developed a method which represents a body within a flow field via a forcing function added to the governing equations. His immersed boundary method has been successfully implemented into moving boundary problems and extended to an adaptive grid solver. Goldstein et al. [32] used an approach called "virtual boundary method" to solve rigid boundary problem, in which the surface boundary condition is specified by proper forcing function. In this approach, the boundary force is implemented into the neighbor grid points by multiplying Gauss distribution to ensure smooth surface. The virtual boundary method was applied by Saiki and Biringen [73] to simulate the flow past a cylinder using fluid-boundary interaction

method, in which the boundary values were interpolated from grid point values and the boundary forces were extrapolated back to the grid points. All these approaches showed very good results in an incompressible flow calculation. However, the numerical stiffness of the forcing function may result in very small time step, and the finite difference method would require large grid resolution to get a conservative solution. Steinhoff et al. [22, 23, 78-80, 92-94] developed the vorticity confinement method, which provides efficient viscous solution in a vortex dominant flow field without numerical dissipation. This method is to treat vertical layers and filaments as "solitary waves", and to track it using discrete evolution equation in a uniformly spaced Cartesian grid. However, this approach is limited to a subsonic flow, and it has a difficulty in unstructured grid system due to dependency of confinement parameters on grid size. Although this technique allows the use of much coarser grids to model high Reynolds number flow that has compact vortices, it does not capture any of the details of the interior of the vertical regions as it only models these regions as thin lines [55].

With regard to turbulence modeling, two equation turbulence modeling was tested for Cartesian grid solver using adaptive boundary cell by Wang et al [89, 90]. As shown in Figure 6, their grid shape is very close to that of hybrid grid, which cannot take the advantage of Cartesian grid topology. To date, turbulent modeling using zero, one or two equation model has not been tested in an immersed Cartesian grid, since it is very difficult to treat the non-body-fitted boundary cells. In this context, boundary cells are those Cartesian cells that the solid walls intersect. Enforcing the conventional turbulent wall boundary condition on such boundary cells may result in non-physical fluctuation of flow variables or instability without special treatment of surface boundary cells, since non-body-fitted the distance from

the cell centers to the wall fluctuates sharply. It is for this reason that several researchers skip two equation turbulent model and jump directly to LES solver [35, 83, 87] that resolves viscous sublayer, buffer layer and log-law layer using MPI with very large memory requirements.

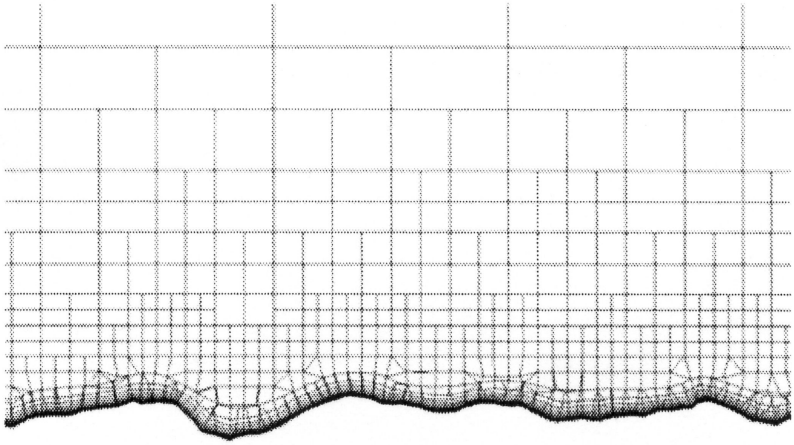


Figure 6 : Viscous-adaptive grid from Ref. [90]

1.2.3 WALL FUNCTION APPROACH

To predict high Reynolds number turbulent boundary layer flows correctly through the viscous sublayer, buffer layer, and outer layer regions, a large number of grid points are required inside the viscous sublayer with near wall value of $y^+ \approx 1$ [5, 10, 26, 44, 74]. Using an explicit unstructured Cartesian grid solver, three-dimensional flow calculation requires extreme number of cells and iterations to get a proper solution. The wall function approach using two equation turbulence modeling, called the standard $k-\varepsilon$ model, introduced by Launder and Spalding [49], is the most grid efficient method to date in the calculation of

RANS equations, eliminating the fine mesh near wall. Although applying a wall function is efficient, it may not be expected to identify the separation point as accurately as high grid resolution turbulence models (such as low Reynolds $k-\varepsilon$ model) can. However, wall function approaches do allow for separation and many researchers have shown that the wall function approach can be highly successful in the calculation of separated flow. Viegas et al.[88] computed the axisymmetric transonic shock-wave/boundary-layer interaction and supersonic flow on deflected flap, and successfully captured the separation phenomena in a compressible flow. Holmes and Connell [37] used the unstructured grid solver with wall function boundary condition to calculate the separation occurred in cascade. The separation in the step flow was simulated by Kwon and Hah [47] in 1995, and showed the well predicted solutions. Mohammadi and Pironneau [59] also applied the standard $k-\varepsilon$ model with the wall function approach to calculate the flows over a backward step and cylinder, and successfully captured the separation as well as the unsteadiness.

The standard $k-\varepsilon$ model with wall function approach is widely used in structured grid solver as well as tetrahedral type unstructured grid solvers [39, 40, 47, 52, 56]. In an immersed Cartesian grid system, it is, however, very difficult to apply the wall function approach and has not done before. The traditional wall function approaches use either slip-wall or no-slip wall conditions, which requires smooth variation of grid cell distance from the wall to the flow cell center. The existence of cell centers located inside of the wall boundary has previously not been considered in such wall functions. However, the boundary cell centers in an immersed Cartesian grid are non-body-fitted so that certain cell centers are located inside of wall boundary. When the no-slip wall boundary condition is employed in

the immersed Cartesian grid solver, non-physical fluctuations of primitive variables near the wall would be induced. Figure 7 shows the computed pressure coefficient over NACA 0012 airfoil when the conventional no-slip wall boundary condition is imposed with wall function approach in immersed Cartesian grid. The non-physical fluctuations of pressure may result in very large adverse pressure gradient and separation, which distorts the whole computational result. Applying slip-wall boundary condition would give non-physical flow acceleration near wall in the transient solution, and require large number of iterations to get the converged solution. To circumvent the problem, many RANS solvers initiate the RANS calculation from the laminar solution, which demands additional computational time.

Another difficulty in applying wall function is the complicated coordinate transformation that is required to integrate the viscous flux. The wall shear stress is calculated from the law of the wall in the body-fitted coordinate. To perform the volume integration, computed shear stress should be transformed to the Cartesian coordinate. This increases the computational cost and attenuates the advantage of Cartesian grid solver.

The conventional wall function approach has a limitation that the closest flow cell center from the wall should be located at $y^+ < 150$. Although the law of the wall shows excellent agreement with various experiments even after, for $y^+ > 300$, the outer law commences, most of the RANS solver requires $y^+ < 80$ for proper turbulence modeling [75, 95]. This condition still needs too many cells to use for multi-geometry or multiple flight condition analysis and design. It is, therefore, strongly required to develop a new boundary condition using wall function approach, which can remove non-physical fluctuation and coordinate transformation in an immersed Cartesian grid.

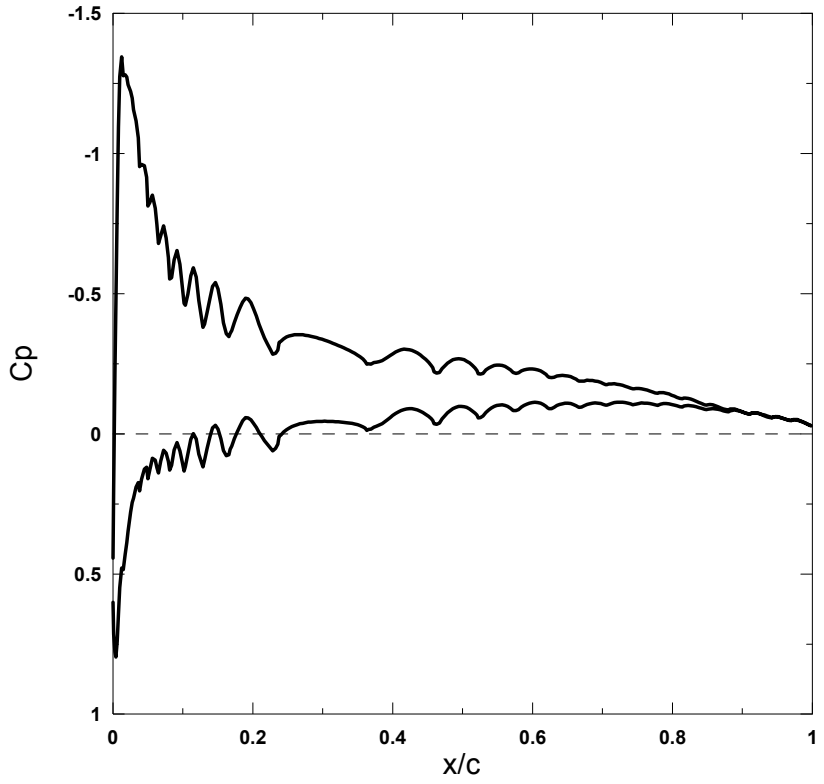


Figure 7 : Non-physical fluctuation of pressure over NACA 0012 airfoil using conventional no-slip wall boundary condition with wall function

1.3 RESEARCH OBJECTIVES

The first objective is to develop an efficient viscous methodology based on the unstructured Cartesian grid technique using an immersed surface boundary cell. The wall boundary condition using ghost cell approach is employed to the existing Navier-Stokes Cartesian grid solver, NASCART-GT. In this approach, the boundary conditions are modified from the method introduced by Marshall and Ruffin [53, 55]. For proper calculation

of high Reynolds number flow, the standard $k-\varepsilon$ model by Launder and Spalding [49] is applied to the solver. To save the computer memory and CPU time in the calculation of turbulent boundary layer, a wall function approach is employed. As mentioned above, applying the wall function approach in an immersed Cartesian grid solver is very challenging, and never done before. The traditional wall function methods are not applicable due to the problems of stability, accuracy and grid efficiency. The research objectives include the development of a new turbulent wall boundary condition that can solve these problems in the Cartesian grid solver, for rotorcraft applications.

The second objective is to develop a CFD tool to model the rotor-fuselage interaction. In order to model the rotor, the boundary condition type actuator disk approach is employed in the developed Cartesian grid solver. Unlike multiple overlapping structured grid topologies, the flow field around complex rotor-fuselage configurations can be modeled as a simple single block grid with unstructured meshes. The three-dimensional Euler calculation is performed over the GT rotor and the ROBIN configuration for the validation of the actuator disk model. The RANS calculation is performed over the GT rotor and the ROBIN rotorcraft model and compared with the Euler solution and experiments to test the capability of capturing viscous phenomena caused by the rotor-fuselage interaction.

CHAPTER II

NUMERICAL FLOW SOLVER

NASCART-GT is an unsteady, three-dimensional Cartesian grid solver of the RANS (Reynolds Averaged Navier-Stokes) equation set. The RANS equations are solved using Roe's approximate Riemann solvers coupled with a MUSCL scheme for inviscid fluxes and traditional finite differencing of the viscous terms. For turbulent flow simulation, the standard $k-\varepsilon$ model by Launder and Spalding [49] is employed. The time integration is performed using a Hancock two-stage scheme which is second order accurate in time. Different from structured grid solver or other unstructured grid solver, the boundary cells of NASCART-GT contacting solid bodies are not body-fitted. Therefore, it is required to develop special techniques to impose wall boundary conditions. In this chapter, they are described in detail.

2.1 GOVERNING EQUATIONS

NASCART-GT assumes the fluid as calorically perfect gas, which is valid for non-hypersonic flows. The molecular viscosity, μ_l, is calculated from the Sutherland's law.

$$\mu_l = C_1 \frac{T^{3/2}}{T + C_2} \tag{2.1}$$

where $C_1 = 1.458 \times 10^{-6}\ kg/\left(m\ s\ \sqrt{K}\right)$, $C_2 = 110.4\ K$

Molecular Prandtl number (Pr) is assumed to be constant, $\mathrm{Pr} = 0.72$. Thermal conductivity is defined from the relation of Prandtl number, specific heat, and viscosity, $k = \frac{c_p}{\mathrm{Pr}}\mu_l$. The following shows the dimensional form of RANS in Tensor notation.

Continuity: $$\frac{\partial \rho}{\partial t} + \frac{\partial}{\partial x_j}\left(\rho u_j\right) = 0$$

Momentum: $$\frac{\partial}{\partial t}\left(\rho u_i\right) + \frac{\partial}{\partial x_j}\left(\rho u_i u_j + \delta_{ij} p - \tau_{ij}\right) = 0$$

Energy: $$\frac{\partial E_t}{\partial t} + \frac{\partial}{\partial x_j}\left[u_j\left(E_t + p\right) - u_i \tau_{ij} + q_j\right] = 0 \qquad (2.2)$$

k -equation: $$\frac{\partial}{\partial t}\left(\rho k\right) + \frac{\partial}{\partial x_j}\left[\rho u_j k - \left(\mu_l + \frac{\mu_t}{\sigma_k}\right)\frac{\partial k}{\partial x_j}\right] = P - \rho\varepsilon$$

ε -equation: $$\frac{\partial}{\partial t}\left(\rho \varepsilon\right) + \frac{\partial}{\partial x_j}\left[\rho u_j \varepsilon - \left(\mu_l + \frac{\mu_t}{\sigma_\varepsilon}\right)\frac{\partial \varepsilon}{\partial x_j}\right] = \left(C_{\varepsilon 1} P - C_{\varepsilon 2}\rho\varepsilon\right)\frac{\varepsilon}{k}$$

E_t represents the total energy per unit volume defined as

$$E_t = \rho\left(e + \frac{1}{2}u_i u_i + k\right) \qquad (2.3)$$

The pressure is coupled with internal energy by the state equation.

$$p = \left(\gamma - 1\right)\rho e \qquad (2.4)$$

Turbulent kinetic energy, k , and dissipation, ε , are defined as

$$k = \frac{1}{2}\overline{u_i' u_i'} \qquad (2.5)$$

$$\varepsilon = 2\nu\overline{s_{ij} s_{ij}} \qquad (2.6)$$

where

$$s_{ij} = \frac{\partial u_i^{'}}{\partial x_j} + \frac{\partial u_j^{'}}{\partial x_i} - \frac{2}{3}\delta_{ij}\frac{\partial u_k^{'}}{\partial x_k}$$

In equation (2.2), the standard model equation for ε is used instead of the exact equation for ε. This is because ε is best viewed as the energy-flow rate in the energy cascade, and it is determined by the large-scale motions, independent of the viscosity at high Reynolds number. By contrast, the exact equation for ε pertains to processes in the dissipative range. Therefore, in the standard model equation for ε is entirely empirical [68]. The shear stress (τ_{ij}), heat flux (q_i), and turbulent production (P) shown in equation (2.1) are defined as

$$\tau_{ij} = (\mu_l + \mu_t)\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} - \frac{2}{3}\delta_{ij}\frac{\partial u_k}{\partial x_k}\right) - \frac{2}{3}\delta_{ij}\rho k \tag{2.7}$$

where δ_{ij} is Kronecker delta function.

$$q_i = -C_p\left(\frac{\mu_l}{\mathrm{Pr}} + \frac{\mu_t}{\mathrm{Pr}_t}\right)\frac{\partial T}{\partial x_i} \tag{2.8}$$

$$P = \left[\mu_t\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} - \frac{2}{3}\delta_{ij}\frac{\partial u_k}{\partial x_k}\right) - \frac{2}{3}\delta_{ij}\rho k\right]\frac{\partial u_i}{\partial x_j} \tag{2.9}$$

The standard k-ε model by Launder and Spalding defines turbulent viscosity and constants as

$$\mu_t = C_\mu \rho \frac{k^2}{\varepsilon} \tag{2.10}$$

$$C_\mu = 0.09, \quad C_{\varepsilon 1} = 1.44, \quad C_{\varepsilon 2} = 1.92, \quad \sigma_k = 1.0, \quad \sigma_\varepsilon = 1.3$$

All the variables shown above are nondimensionalized by the reference length and freestream variables. The advantage in doing this is that the characteristic parameters can be varied

independently. In addition, by nondimensionalizing the equations, the flow variables are normalized so that their values fall between certain prescribed limits. This will reduce numerical errors [81].

$$x_i^* = \frac{x_i}{L}, \qquad t^* = \frac{t}{L/V_\infty}, \qquad u_i^* = \frac{u_i}{V_\infty}, \qquad u_i'^* = \frac{u_i'}{V_\infty}$$

$$\mu_l^* = \frac{\mu_l}{\mu_\infty}, \qquad \mu_t^* = \frac{\mu_t}{\mu_\infty}, \qquad \rho^* = \frac{\rho}{\rho_\infty}, \qquad p^* = \frac{p}{\rho_\infty V_\infty^2} \tag{2.11}$$

$$T^* = \frac{T}{T_\infty}, \qquad e^* = \frac{e}{V_\infty^2}, \qquad k^* = \frac{k}{k_\infty}, \qquad \varepsilon^* = \frac{\varepsilon}{\varepsilon_\infty}$$

where the nondimensional variables are denoted by an asterisk, freestream conditions are denoted by ∞, and L is the reference length used in the Reynolds number.

$$\mathrm{Re}_L = \frac{\rho_\infty V_\infty L}{\mu_\infty} \tag{2.12}$$

When the nondimensionalizing procedure is applied to the RANS equation set in equation (2.2), the following equations are obtained.

Continuity: $$\frac{\partial \rho^*}{\partial t^*} + \frac{\partial}{\partial x_j^*}\left(\rho^* u_j^*\right) = 0$$

Momentum: $$\frac{\partial}{\partial t^*}\left(\rho^* u_i^*\right) + \frac{\partial}{\partial x_j^*}\left(\rho^* u_i^* u_j^* + \delta_{ij} p^* - \tau_{ij}^*\right) = 0$$

Energy: $$\frac{\partial E_t^*}{\partial t^*} + \frac{\partial}{\partial x_j^*}\left[u_j^*\left(E_t^* + p^*\right) - u_i^* \tau_{ij}^* + q_j^*\right] = 0 \tag{2.13}$$

k -equation:

$$\frac{\partial}{\partial t^*}\left(\rho^* k^*\right)+\frac{\partial}{\partial x_j^*}\left[\rho^* u_j^* k^* - \frac{1}{\mathrm{Re}_L}\left(\mu_l^* + \frac{\mu_t^*}{\sigma_k}\right)\frac{\partial k^*}{\partial x_j^*}\right] = P^* - \frac{\varepsilon_\infty L}{k_\infty V_\infty}\rho^* \varepsilon^*$$

ε -equation:

$$\frac{\partial}{\partial t^*}\left(\rho^* \varepsilon^*\right)+\frac{\partial}{\partial x_j^*}\left[\rho^* u_j^* \varepsilon^* - \frac{1}{\mathrm{Re}_L}\left(\mu_l^* + \frac{\mu_t^*}{\sigma_\varepsilon}\right)\frac{\partial \varepsilon^*}{\partial x_j^*}\right] = \left(C_{\varepsilon 1} P^* - C_{\varepsilon 2}\frac{\varepsilon_\infty L}{k_\infty V_\infty}\rho^* \varepsilon^*\right)\frac{\varepsilon^*}{k^*}$$

The nondimensional variables of total energy, shear stress, heat flux, and turbulent production terms are defined as

$$E_t^* = \rho^*\left(e^* + \frac{1}{2}u_i^* u_i^* + \frac{k_\infty}{V_\infty^2}k^*\right) \tag{2.14}$$

$$\tau_{ij}^* = \frac{1}{\mathrm{Re}_L}\left(\mu_l^* + \mu_t^*\right)\left(\frac{\partial u_i^*}{\partial x_j^*} + \frac{\partial u_j^*}{\partial x_i^*} - \frac{2}{3}\delta_{ij}\frac{\partial u_k^*}{\partial x_k^*}\right) - \frac{2}{3}\frac{k_\infty}{V_\infty^2}\delta_{ij}\rho^* k^* \tag{2.15}$$

$$q_i^* = -\frac{1}{(\gamma-1)M_\infty^2\,\mathrm{Re}_L}\left(\frac{\mu_l^*}{\mathrm{Pr}} + \frac{\mu_t^*}{\mathrm{Pr}_t}\right)\frac{\partial T^*}{\partial x_i^*} \tag{2.16}$$

$$P^* = \left[\frac{V_\infty^2}{k_\infty}\frac{\mu_t^*}{\mathrm{Re}_L}\left(\frac{\partial u_i^*}{\partial x_j^*} + \frac{\partial u_j^*}{\partial x_i^*} - \frac{2}{3}\delta_{ij}\frac{\partial u_k^*}{\partial x_k^*}\right) - \frac{2}{3}\delta_{ij}\rho^* k^*\right]\frac{\partial u_i^*}{\partial x_j^*} \tag{2.17}$$

where freestream Mach number is

$$M_\infty = \frac{V_\infty}{\sqrt{\gamma RT_\infty}} \tag{2.18}$$

The state equation is also nondimensionalized as

$$p^* = (\gamma-1)\rho^* e^* \tag{2.19}$$

Nondimensional turbulent viscosity is derived as

$$\mu_t^* = C_\mu \, \mathrm{Re}_T \, \rho^* \frac{k^{*2}}{\varepsilon^*} \tag{2.20}$$

where Re_T is turbulent Reynolds number defined as

$$\mathrm{Re}_T = \frac{\rho_\infty k_\infty^2}{\varepsilon_\infty \mu_\infty} \tag{2.21}$$

The nondimensional governing equation set and collateral relations in tensor form is to be modified to the state vector form in Cartesian coordinate.

$$\frac{\partial \boldsymbol{U}^*}{\partial t^*} + \frac{\partial \boldsymbol{F}^*}{\partial x^*} + \frac{\partial \boldsymbol{G}^*}{\partial y^*} + \frac{\partial \boldsymbol{H}^*}{\partial z^*} = \boldsymbol{S}^* \tag{2.22}$$

where

$$\boldsymbol{U}^* = \begin{bmatrix} \rho^* \\ \rho^* u^* \\ \rho^* v^* \\ \rho^* w^* \\ E_t^* \\ \rho^* k^* \\ \rho^* \varepsilon^* \end{bmatrix}$$

$$\boldsymbol{F}^* = \boldsymbol{F}_{\boldsymbol{I}}^* - \boldsymbol{F}_{\boldsymbol{V}}^*, \qquad \boldsymbol{F}_I^* = \begin{bmatrix} \rho^* u^* \\ \rho^* u^{*2} + p^* \\ \rho^* u^* v^* \\ \rho^* u^* w^* \\ u^* \left(E_t^* + p \right) \\ \rho^* u^* k^* \\ \rho^* u^* \varepsilon^* \end{bmatrix} \qquad \boldsymbol{F}_V^* = \begin{bmatrix} 0 \\ \tau_{xx}^* \\ \tau_{xy}^* \\ \tau_{xz}^* \\ u^* \tau_{xx}^* + v^* \tau_{xy}^* + w^* \tau_{xz}^* - q_x^* \\ \dfrac{1}{\mathrm{Re}_L} \left(\mu_l^* + \dfrac{\mu_t^*}{\sigma_k} \right) \dfrac{\partial k^*}{\partial x^*} \\ \dfrac{1}{\mathrm{Re}_L} \left(\mu_l^* + \dfrac{\mu_t^*}{\sigma_\varepsilon} \right) \dfrac{\partial \varepsilon^*}{\partial x^*} \end{bmatrix}$$

$$\boldsymbol{G}^* = \boldsymbol{G}_{\boldsymbol{I}}^* - \boldsymbol{G}_{\boldsymbol{V}}^*, \qquad \boldsymbol{G}_I^* = \begin{bmatrix} \rho^* v^* \\ \rho^* u^* v^* \\ \rho^* v^{*2} + p^* \\ \rho^* v^* w^* \\ v^* \left(E_t^* + p \right) \\ \rho^* v^* k^* \\ \rho^* v^* \varepsilon^* \end{bmatrix} \qquad \boldsymbol{G}_V^* = \begin{bmatrix} 0 \\ \tau_{xy}^* \\ \tau_{yy}^* \\ \tau_{yz}^* \\ u^* \tau_{xy}^* + v^* \tau_{yy}^* + w^* \tau_{yz}^* - q_y^* \\ \dfrac{1}{\mathrm{Re}_L} \left(\mu_l^* + \dfrac{\mu_t^*}{\sigma_k} \right) \dfrac{\partial k^*}{\partial y^*} \\ \dfrac{1}{\mathrm{Re}_L} \left(\mu_l^* + \dfrac{\mu_t^*}{\sigma_\varepsilon} \right) \dfrac{\partial \varepsilon^*}{\partial y^*} \end{bmatrix}$$

$$\boldsymbol{H}^* = \boldsymbol{H}_I^* - \boldsymbol{H}_V^*, \qquad \boldsymbol{H}_I^* = \begin{bmatrix} \rho^* w^* \\ \rho^* u^* w^* \\ \rho^* v^* w^* \\ \rho^* w^{*2} + p^* \\ w^*\left(E_t^* + p\right) \\ \rho^* w^* k^* \\ \rho^* w^* \varepsilon^* \end{bmatrix} \qquad \boldsymbol{H}_V^* = \begin{bmatrix} 0 \\ \tau_{xz}^* \\ \tau_{yz}^* \\ \tau_{zz}^* \\ u^*\tau_{xz}^* + v^*\tau_{yz}^* + w^*\tau_{zz}^* - q_z^* \\ \dfrac{1}{\mathrm{Re}_L}\left(\mu_l^* + \dfrac{\mu_t^*}{\sigma_k}\right)\dfrac{\partial k^*}{\partial z^*} \\ \dfrac{1}{\mathrm{Re}_L}\left(\mu_l^* + \dfrac{\mu_t^*}{\sigma_\varepsilon}\right)\dfrac{\partial \varepsilon^*}{\partial z^*} \end{bmatrix}$$

$$\boldsymbol{S}^* = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ P^* - \dfrac{\varepsilon_\infty L}{V_\infty k_\infty}\rho^*\varepsilon^* \\ \left(C_{\varepsilon 1}P^* - C_{\varepsilon 2}\dfrac{\varepsilon_\infty L}{V_\infty k_\infty}\rho^*\varepsilon^*\right)\dfrac{\varepsilon^*}{k^*} \end{bmatrix}$$

Tensor forms of several nondimensional variables are rephrased as

$$E_t^* = \rho^*\left[\rho^* + \frac{1}{2}\left(u^{*2} + v^{*2} + w^{*2}\right) + \frac{k_\infty}{V_\infty^2}k^*\right] \tag{2.23}$$

$$\tau_{xx}^{*}=\frac{2}{3}\frac{\left(\mu_{l}^{*}+\mu_{t}^{*}\right)}{\mathrm{Re}_{L}}\left(2\frac{\partial u^{*}}{\partial x^{*}}-\frac{\partial v^{*}}{\partial y^{*}}-\frac{\partial w^{*}}{\partial z^{*}}\right)-\frac{2}{3}\frac{k_{\infty}}{V_{\infty}^{2}}\rho^{*}k^{*}$$

$$\tau_{yy}^{*}=\frac{2}{3}\frac{\left(\mu_{l}^{*}+\mu_{t}^{*}\right)}{\mathrm{Re}_{L}}\left(2\frac{\partial v^{*}}{\partial y^{*}}-\frac{\partial u^{*}}{\partial x^{*}}-\frac{\partial w^{*}}{\partial z^{*}}\right)-\frac{2}{3}\frac{k_{\infty}}{V_{\infty}^{2}}\rho^{*}k^{*}$$

$$\tau_{zz}^{*}=\frac{2}{3}\frac{\left(\mu_{l}^{*}+\mu_{t}^{*}\right)}{\mathrm{Re}_{L}}\left(2\frac{\partial w^{*}}{\partial z^{*}}-\frac{\partial u^{*}}{\partial x^{*}}-\frac{\partial v^{*}}{\partial y^{*}}\right)-\frac{2}{3}\frac{k_{\infty}}{V_{\infty}^{2}}\rho^{*}k^{*} \qquad (2.24)$$

$$\tau_{xy}^{*}=\frac{\left(\mu_{l}^{*}+\mu_{t}^{*}\right)}{\mathrm{Re}_{L}}\left(\frac{\partial u^{*}}{\partial y^{*}}+\frac{\partial v^{*}}{\partial x^{*}}\right)$$

$$\tau_{xz}^{*}=\frac{\left(\mu_{l}^{*}+\mu_{t}^{*}\right)}{\mathrm{Re}_{L}}\left(\frac{\partial u^{*}}{\partial z^{*}}+\frac{\partial w^{*}}{\partial x^{*}}\right)$$

$$\tau_{yz}^{*}=\frac{\left(\mu_{l}^{*}+\mu_{t}^{*}\right)}{\mathrm{Re}_{L}}\left(\frac{\partial v^{*}}{\partial z^{*}}+\frac{\partial w^{*}}{\partial y^{*}}\right)$$

$$q_{x}^{*}=-\frac{1}{(\gamma-1)M_{\infty}^{2}\,\mathrm{Re}_{L}}\left(\frac{\mu_{l}^{*}}{\mathrm{Pr}}+\frac{\mu_{t}^{*}}{\mathrm{Pr}_{t}}\right)\frac{\partial T^{*}}{\partial x^{*}}$$

$$q_{y}^{*}=-\frac{1}{(\gamma-1)M_{\infty}^{2}\,\mathrm{Re}_{L}}\left(\frac{\mu_{l}^{*}}{\mathrm{Pr}}+\frac{\mu_{t}^{*}}{\mathrm{Pr}_{t}}\right)\frac{\partial T^{*}}{\partial y^{*}} \qquad (2.25)$$

$$q_{z}^{*}=-\frac{1}{(\gamma-1)M_{\infty}^{2}\,\mathrm{Re}_{L}}\left(\frac{\mu_{l}^{*}}{\mathrm{Pr}}+\frac{\mu_{t}^{*}}{\mathrm{Pr}_{t}}\right)\frac{\partial T^{*}}{\partial z^{*}}$$

$$P^* = \frac{V_\infty^2}{k_\infty}\frac{\mu_t^*}{\mathrm{Re}_L}\left[\frac{2}{3}\left\{\left(\frac{\partial u^*}{\partial x^*}-\frac{\partial v^*}{\partial y^*}\right)^2+\left(\frac{\partial u^*}{\partial x^*}-\frac{\partial w^*}{\partial z^*}\right)^2+\left(\frac{\partial v^*}{\partial y^*}-\frac{\partial w^*}{\partial z^*}\right)^2\right\}\right.$$
$$\left.+\left(\frac{\partial u^*}{\partial y^*}+\frac{\partial v^*}{\partial x^*}\right)^2+\left(\frac{\partial u^*}{\partial z^*}+\frac{\partial w^*}{\partial x^*}\right)^2+\left(\frac{\partial v^*}{\partial z^*}+\frac{\partial w^*}{\partial y^*}\right)^2\right] \qquad (2.26)$$
$$-\frac{2}{3}\rho^* k^*\left(\frac{\partial u^*}{\partial x^*}+\frac{\partial v^*}{\partial y^*}+\frac{\partial w^*}{\partial z^*}\right)$$

From the following section, all flow and fluid properties without asterisk superscripts represent nondimensional values unless they are specified.

2.2 FLUX CALCULATIONS

To calculate the inviscid fluxes, Roe's Approximate Riemann Solver was used with MUSCL (Monotone Upstream-centered Schemes for Conservation Laws) differencing technique for inviscid fluxes [81]. In the governing equation, inviscid and viscous terms were shown. The viscous flux calculations of the flow cells and boundary cells are performed using standard second order central difference approximations. The difference stencil is populated such that at refinement boundaries the differencing still appears as a uniform sized grid, which can be extended to third order accuracy. The next section provides more details on them.

2.2.1 ROE'S APPROXIMATE RIEMANN SOLVER

The following section explains the mechanism of Roe's approximate Riemann solver and its application to RANS equations including turbulent kinetic energy and dissipation transport equations

In computing the solution to nonlinear systems, it is efficient to solve an approximate Riemann problem rather than having to deal with the exact nonlinear iterative scheme. It models the flow phenomena as a collection of local wave propagation between control volumes. Roe's approximate Riemann solver belongs to this class of solution procedures. Roe's method provides a method of calculating the flux across a face of a control volume using the eigenvalues. For x-directional flux,

$$\begin{aligned} &\frac{\partial \boldsymbol{U}}{\partial t}+\frac{\partial \boldsymbol{F}}{\partial x}=\boldsymbol{0} \\ &\frac{\partial \boldsymbol{U}}{\partial t}+\frac{\partial \boldsymbol{F}}{\partial \boldsymbol{U}}\frac{\partial \boldsymbol{U}}{\partial x}=\boldsymbol{0} \\ &\frac{\partial \boldsymbol{U}}{\partial t}+\boldsymbol{A}\frac{\partial \boldsymbol{U}}{\partial x}=\boldsymbol{0} \end{aligned} \qquad (2.27)$$

$$\boldsymbol{U}(x,0)=\begin{cases}\boldsymbol{U}_{\boldsymbol{L}}, & x<0\\ \boldsymbol{U}_{\boldsymbol{R}}, & x>0\end{cases}$$

Roe's linear approximation to the Riemann problem is written

$$\frac{\partial \boldsymbol{U}}{\partial t}+\tilde{\boldsymbol{A}}\frac{\partial \boldsymbol{U}}{\partial x}=\boldsymbol{0} \qquad (2.28)$$

$\tilde{\boldsymbol{A}}$ is Roe's averaged matrix and is assumed to be a constant in this formulation. Recall that $\boldsymbol{A} = \dfrac{\partial \boldsymbol{F}}{\partial \boldsymbol{U}}$ is the Jacobian matrix for the general case. The components of the $\tilde{\boldsymbol{A}}$ matrix are evaluated using averaged values of $\boldsymbol{U}$ at the interface separating the two states, i.e.

$$\tilde{\boldsymbol{A}} = \tilde{\boldsymbol{A}}(\boldsymbol{U}_L, \boldsymbol{U}_R) \tag{2.29}$$

The Roe-averaged matrix $\tilde{\boldsymbol{A}}$ is chosen to satisfy certain conditions, so that a solution of the linear problem becomes an approximate solution of the nonlinear Riemann problem. These conditions include the following.

1. A linear mapping relates the vector space $\boldsymbol{U}$ to the vector space F.
2. As $\boldsymbol{U}_L$ approaches $\boldsymbol{U}_R$, i.e., as an undisturbed state is reached,

   $$\tilde{\boldsymbol{A}}(\boldsymbol{U}_L, \boldsymbol{U}_R) \Rightarrow \boldsymbol{A}$$

   when

   $$\boldsymbol{U}_L \rightarrow \boldsymbol{U}_R \rightarrow \boldsymbol{U}$$

3. For any two values $\boldsymbol{U}_L$, $\boldsymbol{U}_R$, the jump condition across the interface must be correct, i.e.,

   $$\boldsymbol{F}_R - \boldsymbol{F}_L = \tilde{\boldsymbol{A}}(\boldsymbol{U}_R - \boldsymbol{U}_L) \tag{2.30}$$

4. The eigenvalues of $\tilde{\boldsymbol{A}}$ are real and linearly independent.

This system is a hyperbolic that may be diagonalized by writing the constant matrix $\tilde{\boldsymbol{A}}$ as

$$\tilde{\boldsymbol{A}} = \tilde{\boldsymbol{T}} \tilde{\boldsymbol{\Lambda}} \tilde{\boldsymbol{T}}^{-1} \tag{2.31}$$

Defining the vector W as

$$\boldsymbol{W} = \tilde{\boldsymbol{T}}^{-1} \boldsymbol{U} \tag{2.32}$$

leads the linear problem

$$\frac{\partial \boldsymbol{W}}{\partial t} + \tilde{\boldsymbol{\Lambda}} \frac{\partial \boldsymbol{W}}{\partial x} = \boldsymbol{0} \tag{2.33}$$

where the matrix of eigenvalues $\tilde{\boldsymbol{\Lambda}}$ is a diagonal matrix. For a single linear equation, the value of $\boldsymbol{W}$ is constant along the characteristic defined by $dx/dt = \lambda_k$, since this system is expressed as the matrix shown below.

$$\begin{bmatrix} 1 & \lambda_k \\ dt & dx \end{bmatrix} \begin{bmatrix} W_t \\ W_x \end{bmatrix} = \begin{bmatrix} 0 \\ dW \end{bmatrix} \tag{2.34}$$

As each of the waves associated with the eigenvalues of the system is crossed, the values of the dependent variables experience a jump. Consequently, the values of W_k are constant between each pair of waves in the domain. Mathematically,

W_k =const

when

$\lambda_{k-1} \le x/t \le \lambda_k$

Consequently, the value of W and U at any point may be written

$$W_k = W_1 + \sum_{j=2}^{k} \left(W_j - W_{j-1} \right) \tag{2.35}$$

$$U_k = U_1 + \sum_{j=2}^{k} \left(U_j - U_{j-1} \right) \tag{2.36}$$

Finally,

$$F_k = F_1 + \sum_{j=2}^{k} \delta F_j \tag{2.37}$$

where the flux increments are associated with the crossing of each wave in the system.

If the entire wave system is traversed and the left and right states are identified with appropriate subscripts, then

$$\boldsymbol{F}_{\boldsymbol{R}} = \boldsymbol{F}_{\boldsymbol{L}} + \tilde{\boldsymbol{A}}\left(\boldsymbol{U}_{\boldsymbol{R}} - \boldsymbol{U}_{\boldsymbol{L}}\right) \tag{2.38}$$

The $\tilde{\boldsymbol{A}}$ matrix may be split, corresponding to changes that occur across negative and positive waves. Consequently, we may split the calculation of the fluxes into contributions across negative and positive waves to determine appropriate formulas for the cell-face fluxes in the linear Riemann problem.

$$\begin{aligned} \boldsymbol{F}_{i+\frac{1}{2}} &= \boldsymbol{F}_{\boldsymbol{L}} + \tilde{\boldsymbol{A}}^{-}\left(\boldsymbol{U}_{\boldsymbol{R}} - \boldsymbol{U}_{\boldsymbol{L}}\right) \\ \boldsymbol{F}_{i+\frac{1}{2}} &= \boldsymbol{F}_{\boldsymbol{R}} - \tilde{\boldsymbol{A}}^{+}\left(\boldsymbol{U}_{\boldsymbol{R}} - \boldsymbol{U}_{\boldsymbol{L}}\right) \end{aligned} \tag{2.39}$$

Averaging those gives

$$\boldsymbol{F}_{i+\frac{1}{2}} = \frac{\boldsymbol{1}}{\boldsymbol{2}}\left\{\left(\boldsymbol{F}_{\boldsymbol{L}} + \boldsymbol{F}_{\boldsymbol{R}}\right) - \left|\tilde{\boldsymbol{A}}\right|\left(\boldsymbol{U}_{\boldsymbol{R}} - \boldsymbol{U}_{\boldsymbol{L}}\right)\right\} \tag{2.40}$$

where $\left|\tilde{\boldsymbol{A}}\right| = \tilde{\boldsymbol{T}}\left|\tilde{\boldsymbol{\Lambda}}\right|\tilde{\boldsymbol{T}}^{-1}$ and $\left|\tilde{\boldsymbol{\Lambda}}\right|$ is the diagonal matrix whose entries are the absolute values of the eigenvalues. Condition 3 and the subsequent expressions for the interface flux show that the change across any wave depends upon the change in state variables across all waves. The second term on the right hand side of equation (2.40) can be manipulated using summation, i.e.

$$\begin{aligned} \left|\tilde{\boldsymbol{A}}\right|\left(\boldsymbol{U}_{\boldsymbol{R}} - \boldsymbol{U}_{\boldsymbol{L}}\right) &= \tilde{\boldsymbol{T}}\left|\tilde{\boldsymbol{\Lambda}}\right|\tilde{\boldsymbol{T}}^{-1}\left(\boldsymbol{U}_{\boldsymbol{R}} - \boldsymbol{U}_{\boldsymbol{L}}\right) \\ &= \sum_{j=1}^{k} \tilde{\alpha}_j \left|\tilde{\lambda}_j\right| \tilde{\boldsymbol{K}}_j \end{aligned} \tag{2.41}$$

Finally,

$$\boldsymbol{F}_{i+\frac{1}{2}} = \frac{1}{2}\left\{\left(\boldsymbol{F}_L + \boldsymbol{F}_R\right) - \sum_{j=1}^{k} \tilde{\alpha}_j \left|\tilde{\lambda}_j\right| \tilde{\boldsymbol{K}}_j \right\} \tag{2.42}$$

where $\tilde{\lambda}_j$ and $\tilde{\boldsymbol{K}}_j$ are the eigenvalues and corresponding eigenvectors respectively. $\tilde{\alpha}_j$ represents the wave strengths.

Extension of Roe's scheme to the transport equations has been done by Morrison [60]. His method was similar with the existing flux calculation scheme in NASCART-GT, such that it is easy to add the inviscid fluxes of transport equations. The following section summaries it.

For x-directional flux, the eigenvalue is solved as

$$\tilde{\boldsymbol{\lambda}} = \begin{bmatrix} \tilde{\lambda}_1 \\ \tilde{\lambda}_2 \\ \tilde{\lambda}_3 \\ \tilde{\lambda}_4 \\ \tilde{\lambda}_5 \\ \tilde{\lambda}_6 \\ \tilde{\lambda}_7 \end{bmatrix} = \begin{bmatrix} \tilde{u} \\ \tilde{u} \\ \tilde{u} \\ \tilde{u} + \tilde{a} \\ \tilde{u} - \tilde{a} \\ \tilde{u} \\ \tilde{u} \end{bmatrix} \tag{2.43}$$

As a result, the flux vector on the control surface can be expressed in a computationally efficient manner as

$$\boldsymbol{F}_{i+\frac{1}{2}} = \frac{1}{2}\left(\boldsymbol{F}_L + \boldsymbol{F}_R - \left|\Delta \boldsymbol{F}_1\right| - \left|\Delta \boldsymbol{F}_4\right| - \left|\Delta \boldsymbol{F}_5\right|\right) \tag{2.44}$$

where

$$\left|\Delta \boldsymbol{F}_1\right| = \left|\tilde{u}\right| \left\{ \left(\Delta\rho - \frac{\Delta p}{\tilde{a}^2} \right) \begin{bmatrix} 1 \\ \tilde{u} \\ \tilde{v} \\ \tilde{w} \\ \tilde{q}^2/2 + \tilde{k} \\ \tilde{k} \\ \tilde{\varepsilon} \end{bmatrix} + \tilde{\rho} \begin{bmatrix} 0 \\ 0 \\ \Delta v \\ \Delta w \\ \Delta\left(q^2/2\right) - \tilde{u}\Delta u + \Delta k \\ \Delta k \\ \Delta\varepsilon \end{bmatrix} \right\}$$

$\left|\Delta \boldsymbol{F}_1\right|$ includes wave effect associated with eigenvalue $\tilde{u}$, i.e. λ_1, λ_2, λ_3, λ_6, and λ_7.

$$\left|\Delta \boldsymbol{F}_4\right| = \left|\tilde{u} + \tilde{a}\right| \frac{\left(\Delta p + \tilde{\rho}\tilde{a}\Delta u\right)}{2\tilde{a}^2} \begin{bmatrix} 1 \\ \tilde{u} + \tilde{a} \\ \tilde{v} \\ \tilde{w} \\ \tilde{H} + \tilde{a}\tilde{u} \\ \tilde{k} \\ \tilde{\varepsilon} \end{bmatrix}$$

$$\left|\Delta \boldsymbol{F}_5\right| = \left|\tilde{u} - \tilde{a}\right| \frac{\left(\Delta p - \tilde{\rho}\tilde{a}\Delta u\right)}{2\tilde{a}^2} \begin{bmatrix} 1 \\ \tilde{u} - \tilde{a} \\ \tilde{v} \\ \tilde{w} \\ \tilde{H} - \tilde{a}\tilde{u} \\ \tilde{k} \\ \tilde{\varepsilon} \end{bmatrix}$$

where Δ represents the difference of properties between right and left side of the face.

$$\tilde{\rho} = \sqrt{\rho_L \rho_R}$$

$$\tilde{u} = \frac{\rho_L^{1/2} u_L + \rho_R^{1/2} u_R}{\rho_L^{1/2} + \rho_R^{1/2}}$$

$$\tilde{v} = \frac{\rho_L^{1/2} v_L + \rho_R^{1/2} v_R}{\rho_L^{1/2} + \rho_R^{1/2}}$$

$$\tilde{w} = \frac{\rho_L^{1/2} w_L + \rho_R^{1/2} w_R}{\rho_L^{1/2} + \rho_R^{1/2}}$$

$$\tilde{q}^2 = \tilde{u}^2 + \tilde{v}^2 + \tilde{w}^2$$

$$\tilde{H} = \frac{\rho_L^{1/2} H_L + \rho_R^{1/2} H_R}{\rho_L^{1/2} + \rho_R^{1/2}} \tag{2.45}$$

$$H = \text{Total enthalpy per unit mass} = \frac{1}{\rho}\left(E_t + p\right)$$

$$\tilde{k} = \frac{\rho_L^{1/2} k_L + \rho_R^{1/2} k_R}{\rho_L^{1/2} + \rho_R^{1/2}}$$

$$\tilde{\varepsilon} = \frac{\rho_L^{1/2}\varepsilon_L + \rho_R^{1/2}\varepsilon_R}{\rho_L^{1/2} + \rho_R^{1/2}}$$

$$\tilde{a} = \sqrt{(\gamma - 1)\left(\tilde{H} - \frac{\tilde{q}^2}{2} - \tilde{k}\right)}$$

Apply these equations to a Cartesian control volume results in the following formulation, then

$$\hat{\boldsymbol{F}}_{LR} = \frac{1}{2}\left(\hat{\boldsymbol{F}}_{\boldsymbol{L}} + \hat{\boldsymbol{F}}_{\boldsymbol{R}} - \left|\Delta\hat{\boldsymbol{F}}_1\right| - \left|\Delta\hat{\boldsymbol{F}}_4\right| - \left|\Delta\hat{\boldsymbol{F}}_5\right|\right) \tag{2.46}$$

where

$$\left|\Delta\boldsymbol{F}_1\right| = \left|\tilde{\phi}\right|\left\{\left(\Delta\rho - \frac{\Delta p}{\tilde{a}^2}\right)\begin{bmatrix}1\\ \tilde{u}\\ \tilde{v}\\ \tilde{w}\\ \tilde{q}^2/2\\ \tilde{k}\\ \tilde{\varepsilon}\end{bmatrix} + \tilde{\rho}\begin{bmatrix}0\\ \Delta u - n_x\Delta\phi\\ \Delta v - n_y\Delta\phi\\ \Delta w - n_z\Delta\phi\\ \Delta\left(q^2/2\right) - \tilde{\phi}\Delta\phi + \Delta k\\ \Delta k\\ \Delta\varepsilon\end{bmatrix}\right\}$$

$$\left|\Delta\boldsymbol{F}_4\right| = \left|\tilde{\phi} + \tilde{a}\right|\frac{\left(\Delta p + \tilde{\rho}\tilde{a}\Delta\phi\right)}{2\tilde{a}^2}\begin{bmatrix}1\\ \tilde{u} + n_x\tilde{a}\\ \tilde{v} + n_y\tilde{a}\\ \tilde{w} + n_z\tilde{a}\\ \tilde{H} + \tilde{a}\tilde{\phi}\\ \tilde{k}\\ \tilde{\varepsilon}\end{bmatrix}$$

$$\left|\Delta \mathrm{F}_5\right| = \left|\widetilde{\phi} - \widetilde{a}\right| \frac{\left(\Delta p - \widetilde{\rho}\widetilde{a}\Delta\phi\right)}{2\widetilde{a}^2} \begin{bmatrix} 1 \\ \widetilde{u} - n_x\widetilde{a} \\ \widetilde{v} - n_y\widetilde{a} \\ \widetilde{w} - n_z\widetilde{a} \\ \widetilde{H} - \widetilde{a}\widetilde{\phi} \\ \widetilde{k} \\ \widetilde{\varepsilon} \end{bmatrix}$$

For the flux calculation in the x-direction, L/R varies in the x-direction and

$$\hat{\boldsymbol{F}} = \boldsymbol{F}$$

$$\phi = u$$

$$\widetilde{\phi} = \widetilde{u}$$

$$\boldsymbol{n} = \begin{bmatrix} n_x \\ n_y \\ n_z \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

For the flux calculation in the y-direction, L/R varies in the y-direction and

$$\hat{\boldsymbol{F}} = \boldsymbol{G}$$

$$\phi = v$$

$$\widetilde{\phi} = \widetilde{v}$$

$$\boldsymbol{n} = \begin{bmatrix} n_x \\ n_y \\ n_z \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$$

For the flux calculation in the z-direction, L/R varies in the z-direction and

$$\hat{\boldsymbol{F}} = \boldsymbol{H}$$

$$\phi = w$$

$$\widetilde{\phi} = \widetilde{w}$$

$$\boldsymbol{n} = \begin{bmatrix} n_x \\ n_y \\ n_z \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

2.2.2 MUSCL DATA RECONSTRUCTION

In the Riemann or approximate Riemann solvers, a higher-order approximation must be interpreted in terms of flux values at control-volume boundaries. For the variable extrapolation approach, van Leer [51] coined the term "monotone upstream-centered schemes for conservation laws" introducing a piece-wise linear reconstruction of the primitive state variable instead of the piece-wise constant reconstruction used in lower order Godunov schemes. This is referred to as the MUSCL approach, or sometimes MUSCL differencing. The expressions for the right and left interpolations are

$$\begin{aligned} \boldsymbol{W}_{L_{i+\frac{1}{2},j,k}} &= \boldsymbol{W}_{i,j,k} + \frac{\varepsilon_{i,j,k}}{4}\left[(1-\kappa)\left(\boldsymbol{W}_{i,j,k} - \boldsymbol{W}_{i-1,j,k}\right) + (1+\kappa)\left(\boldsymbol{W}_{i+1,j,k} - \boldsymbol{W}_{i,j,k}\right)\right] \\ \boldsymbol{W}_{R_{i+\frac{1}{2},j,k}} &= \boldsymbol{W}_{i+1,j,k} - \frac{\varepsilon_{i,j,k}}{4}\left[(1+\kappa)\left(\boldsymbol{W}_{i+1,j,k} - \boldsymbol{W}_{i,j,k}\right) + (1-\kappa)\left(\boldsymbol{W}_{i+2,j,k} - \boldsymbol{W}_{i+1,j,k}\right)\right] \end{aligned} \tag{2.47}$$

where

$$\boldsymbol{W} = \begin{bmatrix} \rho \\ u \\ v \\ w \\ H \\ k \\ \varepsilon \end{bmatrix}$$

This formulation is convenient to use for uniform grid or transformed unstructured grid, since they have uniform grid spacing. In NASCART-GT, neighbor cells may, however, have different refinement level and multiple levels of cell sizes. To increase the accuracy, it is required to modify the MUSCL differencing such that it can take into account of variable grid spacing. The modified formulation is shown below.

$$\boldsymbol{W}_{L_{i+\frac{1}{2},j,k}} = \boldsymbol{W}_{i,j,k} + \frac{\varepsilon_{i,j,k}\Delta x_{i,j,k}}{2}\left[(1-\kappa)\frac{\boldsymbol{W}_{i,j,k} - \boldsymbol{W}_{i-1,j,k}}{\Delta x_{i,j,k} + \Delta x_{i-1,j,k}} + (1+\kappa)\frac{\boldsymbol{W}_{i+1,j,k} - \boldsymbol{W}_{i,j,k}}{\Delta x_{i+1,j,k} + \Delta x_{i,j,k}}\right]$$

$$\boldsymbol{W}_{R_{i+\frac{1}{2},j,k}} = \boldsymbol{W}_{i+1,j,k} - \frac{\varepsilon_{i,j,k}\Delta x_{i+1,j,k}}{2}\left[(1+\kappa)\frac{\boldsymbol{W}_{i+1,j,k} - \boldsymbol{W}_{i,j,k}}{\Delta x_{i+1,j,k} + \Delta x_{i,j,k}} + (1-\kappa)\frac{\boldsymbol{W}_{i+2,j,k} - \boldsymbol{W}_{i+1,j,k}}{\Delta x_{i+2,j,k} + \Delta x_{i+1,j,k}}\right] \tag{2.48}$$

The reconstructed data can be plugged into a flux reconstruction scheme, such as Roe scheme to produce the inviscid fluxes. Reconstruction of the other 5 faces follows in a similar fashion. The values of MUSCL scheme, $\varepsilon_{i,j,k}$ and κ , are presented in Table 1 with corresponding schemes.

Table 1 : Coefficients of MUSCL scheme

$\varepsilon_{i,j,k}$	κ	Scheme
0	N/A	First order piece-wise constant
1	-1	Second order fully upwind biased scheme
	0	Second order upwind biased scheme
	1/2	Third order upwind biased scheme
	1	Second order central difference scheme

To enhance stability of the calculation and remove non-physical overshoot of flow properties, the monotony of the scheme is introduced via a limiter that sets the data reconstruction to first order in regions of high gradients in density, internal energy and total enthalpy. If the flux difference between a cell center and a cell face is larger than the specified value, $\phi_{\lim}$, the order of accuracy will be reduced. The following shows the formulation on the $(i+1/2, j, k)$ face.

$$\varepsilon_{i+\frac{1}{2},j,k} = \min\left[\varepsilon_{input},\ \max\left(1-\frac{\Delta\phi_{\max}}{\phi_{\lim}},\ 0\right)\right] \tag{2.49}$$

$$\Delta\phi_{\max} = \max\left[\left\{\max\left(\phi_{L_{i+\frac{1}{2},j,k}},\ \phi_{R_{i+\frac{1}{2},j,k}}\right)-\max\left(\phi_{i,j,k},\ \phi_{i+1,j,k}\right)\right\},\right.$$
$$\left.\left\{\min\left(\phi_{L_{i+\frac{1}{2},j,k}},\ \phi_{R_{i+\frac{1}{2},j,k}}\right)-\min\left(\phi_{i,j,k},\ \phi_{i+1,j,k}\right)\right\},0\right]$$

where ϕ represents density, internal energy and total enthalpy, and $\phi_{\lim}$ is set to

$$\phi_{\lim} = 0.05\cdot\min\left(\phi_{i,j,k},\phi_{i+1,j,k}\right) \tag{2.50}$$

The order of accuracy for cells contacting solid body was set to first order due to stability problem. The flux calculation of turbulent kinetic energy and dissipation rate is also set to first order to avoid instability. Solid surface flux did not need to be calculated since the cell center location is independent of solid body surface.

2.2.3 NUMERICAL STENCIL

A numerical stencil is constructed to calculate the inviscid and viscous fluxes in NASCART. It determines the state vectors on the local cell and then performs a non-uniformly-spaced finite difference approximation to calculate the fluxes. With the grid refined in different level, there are three grid configurations possible, a locally uniform grid, a local grid with fine neighbors and a local grid with coarse neighbors.

For a locally uniform grid, no special treatment is required and the state of the neighboring control volumes can be used as it is. To calculate the flux on the flux faces, it is required to find the state vectors of "i-2" through "i+2" for second or higher order scheme. Figure 8 shows an example of non-uniform grid structure. The "o" marks represent the numbered cell centers, the "x" the locations where the state vectors are calculated. With the coarse neighbors, the state vectors $\boldsymbol{U}_{i-1}$ and $\boldsymbol{U}_{i-2}$ in Figure 8 are interpolated from its neighbor cells of cell number 1 using the second order pseudo-Laplacian weighted averaging, which is explained in Appendix A. For a fine neighbor cells like "i+1" in Figure 8, the state vector $\boldsymbol{U}_{i+1}$ is interpolated from cell number 3 and 4. For more refined neighbors as "i+2", $\boldsymbol{U}_{i+2}$ is interpolated from the state vectors at 5 through 9.

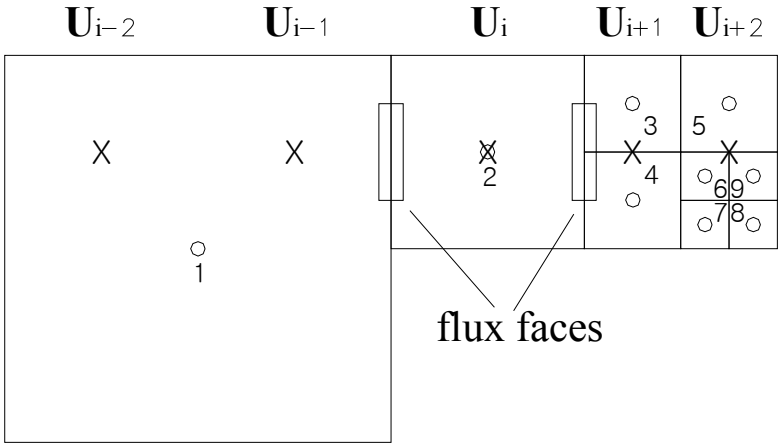


Figure 8 : Numerical stencil calculation

2.3 TIME INTEGRATION AND STABILITY

To provide for efficient, time-accurate solution of the governing equations, a dual time-stepping, multistage scheme has been employed. Hancock's two-stage scheme [84] has been used for time integration as used in compressible flow solver.

It is known that the $k-\varepsilon$ equations are instability prone during the transitory phase of the computations even with an implicit solver, when the leading part of the error surges out of the computational domain, generating large residuals. For stability, G.A. Gerolymos [30] used a limiter to bound k and ε in his implicit solver. Y. Zhao [100] introduced semi-implicit treatment of the source terms of k and ε equations. R.F. Kunz and B. Lakshminarayana [46] studied stability of explicit turbulent solvers. In NASCART-GT, the explicit time step is modified and the limiters of k and ε are used according to the stability

analysis, in order to stabilize the computation and ensure the positivity of k and ε. The following section describes the stability analysis of turbulent k and ε equations.

2.3.1 HANCOCK'S TWO-STAGE SCHEME

Hancock's explicit scheme consists of following two stages.

Step 1 (Predictor):

$$\boldsymbol{U}_{i,j,k}^{n+1/2} = \boldsymbol{U}_{i,j,k}^{n} - \frac{1}{2}\frac{\Delta t_{i,j,k}^{n}}{V_{i,j,k}}\boldsymbol{R}_{i,j,k}^{n} \tag{2.51}$$

Step 2 (Corrector):

$$\boldsymbol{U}_{i,j,k}^{n+1} = \boldsymbol{U}_{i,j,k}^{n} - \frac{\Delta t_{i,j,k}^{n}}{V_{i,j,k}}\boldsymbol{R}_{i,j,k}^{n+1/2} \tag{2.52}$$

where $\boldsymbol{R}^n$ represents the residual vector at time step, n, formulated as:

$$\begin{aligned}\boldsymbol{R}_{i,j,k}^{n} = &\left(\boldsymbol{F}^{n}_{i-\frac{1}{2},j,k} - \boldsymbol{F}^{n}_{i+\frac{1}{2},j,k}\right)A_{1_{i,j,k}} + \left(\boldsymbol{G}^{n}_{i,j-\frac{1}{2},k} - \boldsymbol{G}^{n}_{i,j+\frac{1}{2},k}\right)A_{2_{i,j,k}} + \left(\boldsymbol{H}^{n}_{i,j,k-\frac{1}{2}} - \boldsymbol{H}^{n}_{i,j,k+\frac{1}{2}}\right)A_{3_{i,j,k}} \\ &- \boldsymbol{S}_{i,j,k}^{n}V_{i,j,k}\end{aligned} \tag{2.53}$$

A_1, A_2, and A_3 denote the cell areas perpendicular to x-, y-, and z- axis, respectively.

V is the nondimensional cell volume and Δt^n represents the pseudo-time step calculated as following. For inviscid flows,

$$\Delta t_{i,j,k}^{n} = CFL \cdot \min\left[\frac{\Delta x_{i,j,k}}{\left|u_{i,j,k}\right| + c_{i,j,k}}, \frac{\Delta y_{i,j,k}}{\left|v_{i,j,k}\right| + c_{i,j,k}}, \frac{\Delta z_{i,j,k}}{\left|w_{i,j,k}\right| + c_{i,j,k}}\right] \tag{2.54}$$

For viscous flows,

$$\Delta t^n_{i,j,k} = CFL \cdot \min\left[\frac{\Delta x_{i,j,k}}{\left(\left|u_{i,j,k}\right| + c_{i,j,k}\right)\left(1 + 2/\mathrm{Re}_{\Delta x}\right)}, \frac{\Delta y_{i,j,k}}{\left(\left|v_{i,j,k}\right| + c_{i,j,k}\right)\left(1 + 2/\mathrm{Re}_{\Delta y}\right)}, \frac{\Delta z_{i,j,k}}{\left(\left|w_{i,j,k}\right| + c_{i,j,k}\right)\left(1 + 2/\mathrm{Re}_{\Delta z}\right)} \right] \tag{2.55}$$

where c is the local speed of sound, and the cell Reynolds number is

$$\mathrm{Re}_{\Delta x} = \left|u_{i,j,k}\right| \Delta x_{i,j,k} / \nu_{i,j,k} . \tag{2.56}$$

For turbulent flows, the naïve use of the local time step size determined for the mean flow Navier-Stokes equations lead to unrealistic values of k and ε, and instability. Therefore, the time step calculation is modified to

$$\Delta t^n_{i,j,k} = \min\left[\Delta t^n_{i,j,k}\Big|_{viscous}, (2 - C_{\varepsilon 2}) \frac{k_\infty V_\infty}{\varepsilon_\infty L} \frac{k_{i,j,k}}{\varepsilon_{i,j,k}}, \frac{CFL \cdot k_{i,j,k}}{\left|R_k / V_{i,j,k}\right|}, \frac{CFL \cdot \varepsilon_{i,j,k}}{\left|R_\varepsilon / V_{i,j,k}\right|} \right] \tag{2.57}$$

where $\Delta t^n\big|_{viscous}$ is time step calculated from viscous flow analysis in equation (2.55). R_k and R_ε denote the residuals of k and ε equations, respectively, defined in equation (2.53). The modified time step calculation in equation (2.57) is derived from the stability analysis of k and ε equations, which is described in the next section. For steady flow, the local time stepping has been employed to accelerate the calculation.

2.3.2 TURBULENT STABILITY ANALYSIS

Physically, the turbulent kinetic energy and dissipation rate should be positive. It is known that the negative parts of the source terms cause instability with an explicit scheme. To ensure positivity of turbulent quantities, it is required to perform a full stability analysis of k and ε equations. However, a full analysis of turbulent quantities is not trivial due to the

complexity of convective and source terms. Instead, Zhao [100] introduced the analysis of asymptotic behavior of the turbulent quantities. In flow regions where all flow quantities are uniform and turbulent is isotropic (e.g. outside of boundary layer, farfield, etc.), there is no gradient exist. Then, k and ε equations (2.2) are reduced to the following. Note that all flow properties shown in this section are dimensional values.

$$\frac{dk}{dt} = -\varepsilon \tag{2.58}$$

$$\frac{d\varepsilon}{dt} = -C_{\varepsilon 2}\frac{\varepsilon^2}{k} \tag{2.59}$$

The solution of the equations shows an asymptotic relation:

$$k \approx t^{-1/(C_{\varepsilon 2}-1)} \tag{2.60}$$

$$\varepsilon \approx t^{-C_{\varepsilon 2}/(C_{\varepsilon 2}-1)} \tag{2.61}$$

$$\frac{k^2}{\varepsilon} \approx t^{(2-C_{\varepsilon 2})/(C_{\varepsilon 2}-1)} \tag{2.62}$$

When an explicit scheme is used, then

$$k^{n+1} = k^n\left(1-\Delta t\frac{\varepsilon^n}{k^n}\right) \tag{2.63}$$

$$\varepsilon^{n+1} = \varepsilon^n\left(1-\Delta t C_{\varepsilon 2}\frac{\varepsilon^n}{k^n}\right) \tag{2.64}$$

Since $1 < C_{\varepsilon 2} < 2$, the following condition must be satisfied to ensure positivity of the turbulent quantities:

$$\Delta t < \frac{k^n}{C_{\varepsilon 2}\varepsilon^n} \tag{2.65}$$

According to equation (2.62), μ_t is decreasing as t increases, and

$$\frac{\left(k^{n+1}\right)^2}{\varepsilon^{n+1}} = \frac{\left(k^n\right)^2}{\varepsilon^n} \frac{\left(1-\Delta t\varepsilon^n / k^n\right)^2}{1-\Delta t C_{\varepsilon 2}\varepsilon^n / k^n} \tag{2.66}$$

Therefore, the following holds

$$\frac{\left(1-\Delta t\varepsilon^n / k^n\right)^2}{1-\Delta t C_{\varepsilon 2}\varepsilon^n / k^n} < 1 \tag{2.67}$$

As a result, we have

$$\Delta t < \left(2 - C_{\varepsilon 2}\right)\frac{k^n}{\varepsilon^n} \tag{2.68}$$

For a non-uniform velocity field or non-isotropic turbulent region, the time step can be restricted directly from residual quantities. Suppose ϕ represents k or ε. In order for ϕ to be positive,

$$\phi^{n+1} = \phi^n + \Delta\phi > 0 \tag{2.69}$$

where $\Delta\phi$ is the change of ϕ within a stage. The above equation is equivalent to

$$\frac{\Delta\phi}{\phi^n} > -1$$

A more stringent condition is

$$\left|\frac{\Delta\phi}{\phi^n}\right| = \alpha < 1 \tag{2.70}$$

Multiplying $V/\Delta t$ gives

$$\left|\frac{\Delta\phi}{\phi^n}\frac{V}{\Delta t}\right| = \alpha\frac{V}{\Delta t} \tag{2.71}$$

Since $R_\phi = \frac{V}{\Delta t}\Delta\phi$ and ϕ^n is positive, the time step size is obtained as

$$\Delta t = \alpha \frac{\phi^n}{\left|R_\phi / V\right|} \tag{2.72}$$

In NASCART-GT, the relaxation factor, α, is replaced with CFL number for simplicity. It is reasonable to use CFL number, since it is always positive and less than unity for an explicit solver. As a result,

$$\Delta t_{k,\varepsilon} = \min\left[\left(2 - C_{\varepsilon 2}\right)\frac{k}{\varepsilon}, \frac{CFL \cdot k}{\left|R_k / V\right|}, \frac{CFL \cdot \varepsilon}{\left|R_\varepsilon / V\right|}\right] \tag{2.73}$$

Nondimensionalizing equation (2.73) gives equation (2.57). Nevertheless, simply applying equation (2.57) can result in very small value of time step size when k and ε approach to zero near uniform flow field and isotropic turbulence region. To prevent it, k and ε are bounded by the following limiter as described in references [30, 46, 47].

$$\begin{aligned} &\rho k \geq K_k \rho_\infty k_\infty > 0 \\ &\rho\varepsilon \geq K_\varepsilon \rho_\infty \varepsilon_\infty > 0 \\ &10P \geq \rho\varepsilon \geq 0.1P \end{aligned} \tag{2.74}$$

where $K_k = 0.01 \sim 0.0001$ and $K_\varepsilon = 0.0001$. In NASCART-GT, the values are set to 0.0001. Equation (2.74) imposes $\rho\varepsilon$ to be of the same order of magnitude as P so as to avoid instabilities near the boundary layer edge.

2.4 GRID REFINEMENT BASED ON SOLUTION ADAPTION

The solution adaption methodology using the velocity divergence is discussed by Tu [84] and Marshall [55], where for each control volume, the velocity divergence is scaled by a characteristic length of the control volume to obtain a measure of the changing flow properties from cell to cell. The approach is expanded to vorticity and gradients of turbulent kinetic energy and dissipation.

$$\tau_{\phi_{i,j,k}} = \phi \, l_{i,j,k}^{3/2} \tag{2.75}$$

where l is the cube-root of the cell volume, and the definition of ϕ is shown in Table 2.

Next the root-mean-square is calculated over the entire computational domain to get a reference value, σ_ϕ

$$\sigma_\phi = \frac{1}{N} \sum_{i=1}^{N} \tau_\phi^2 \tag{2.76}$$

where N is the total number of active cells. Finally cells are flagged for coarsening and refinement if the following conditions satisfied.

$$\begin{aligned} \tau_{\phi_{i,j,k}} &< \kappa_c \sigma_\phi \quad \text{Coarsen} \\ \tau_{\phi_{i,j,k}} &> \kappa_r \sigma_\phi \quad \text{Refine} \end{aligned} \tag{2.77}$$

where κ_c and κ_r are threshold values for coarsening and refining, respectively. They are taken to be 0.3 and 1.0, respectively.

Table 2 : Solution adaption parameters

Solution adaption parameter	Definition of ϕ
Velocity divergence	$\left\|\nabla \bullet \boldsymbol{v}_{i,j,k}\right\|$
Vorticity	$\left\|\nabla \times \boldsymbol{v}_{i,j,k}\right\|$
Turbulent kinetic energy gradient	$\left\|\nabla k_{i,j,k}\right\|$
Turbulent dissipation energy gradient	$\left\|\nabla \varepsilon_{i,j,k}\right\|$

The new state vectors of a coarsened cell are averaged from the old cells before coarsening. Those of a refined cell are interpolated from the old cell and its old neighbor cells in order to improve stability and convergence of the computation. If the uniform state vectors are set for a refined cell, this can cause instability and poor convergence due to the significantly large viscous flux caused by large eddy viscosity in the boundary layer.

C H A P T E R I I I

BOUNDARY TREATMENT

The computational domain boundaries consist of solid wall, inflow and outflow boundaries. On each boundary, five variables should be specified for inviscid and laminar flow solver. In turbulent flow, two additional variables are specified, i.e. turbulent kinetic energy and dissipation. Solid wall boundary condition is enforced using the ghost cell approach, which specifies pressure, temperature, and velocity components including turbulent properties. For the wall boundary condition of RANS solver, the wall function approach is used to reduce required computer memory. For outflow and inflow boundaries, characteristic boundary condition or simple extrapolation is applied to specify five or seven Riemann invariants. All flow properties shown in this section are dimensional values.

3.1 INVISCID AND LAMINAR WALL BOUNDARY CONDITIONS

The solid surface treatment requires the decomposition of the control volume velocities into surface oriented directions. A local coordinate system is defined such that η is normal to the surface, and ξ and ζ are perpendicular to each other and are along the surface in the order of a right-handed orthogonal coordinate. ξ is also along the tangential velocity components near wall. With the assumption of invariant quantities on the surface (thin-layer

Navier-Stokes approximations to temperature filed), the $\frac{\partial}{\partial \xi}$ and $\frac{\partial}{\partial \zeta}$ terms shown form applying the chain-rule are zeros. Now, the transformation reduces to

$$\frac{\partial}{\partial x} = \frac{\partial \eta}{\partial x}\frac{\partial}{\partial \eta} = n_x \frac{\partial}{\partial \eta}$$

$$\frac{\partial}{\partial y} = \frac{\partial \eta}{\partial y}\frac{\partial}{\partial \eta} = n_y \frac{\partial}{\partial \eta} \tag{3.1}$$

$$\frac{\partial}{\partial z} = \frac{\partial \eta}{\partial z}\frac{\partial}{\partial \eta} = n_z \frac{\partial}{\partial \eta}$$

where n_x, n_y and n_z denote the normal vector from the surface to the flow field. A standard vector formula is applied to find these values.

$$n_x = \frac{\boldsymbol{a}\cdot \boldsymbol{x}_c + d}{\boldsymbol{a}\cdot \boldsymbol{a}} a_1$$

$$n_y = \frac{\boldsymbol{a}\cdot \boldsymbol{x}_c + d}{\boldsymbol{a}\cdot \boldsymbol{a}} a_2 \tag{3.2}$$

$$n_z = \frac{\boldsymbol{a}\cdot \boldsymbol{x}_c + d}{\boldsymbol{a}\cdot \boldsymbol{a}} a_3$$

where $\boldsymbol{a}\cdot \boldsymbol{x} + d = 0$ is the equation of the surface with $\boldsymbol{a} = a_1 \boldsymbol{i} + a_2 \boldsymbol{j} + a_3 \boldsymbol{k}$, and $\boldsymbol{x}_c$ is the cell center. The surface normal vector, $\boldsymbol{a}$, is calculated using Cramer's rule from the node points of the panel.

Instead of directly specifying boundary conditions on a wall, NASCART-GT enforces primitive variables on the ghost cells such as 15 and 16 in Figure 9, which act as wall boundaries. The state vector of cells touching the solid wall (called boundary cell, for example, 11, 12 and 14 in Figure 9) is found using time integration as the same manner with

flow cells. The point B and D represent the reference points of cell 16 and 15 respectively. The location of reference point is determined as following. First, find the closest point on a body panel from the ghost cell center, i.e. point A or C. Then, extend the line connecting the ghost cell center and the closest point on the wall panel to the extent of predetermined length, δ_r. In current study, δ_r is set to the length of boundary cell diagonal. The primitive variables at a reference point are interpolated from the primitive variables of the 3 closest neighbor cell centers using the linear least square interpolation, described in Appendix A. After finding primitive variables at the reference point, the variables at the ghost cell center are calculated using linear extrapolation.

Inviscid Wall

For pressure, the condition $\frac{\partial p}{\partial \eta} = 0$ is satisfied by setting the pressures on the wall and the ghost cell center are identical to that at the reference cell, i.e. $p_{ref} = p_w = p_g$, where the subscripts *ref*, *w* and *g* denote reference point, wall and ghost cell center, respectively. Adiabatic wall boundary condition is fulfilled when $\frac{\partial T}{\partial \eta} = 0$ is enforced as the same manner with the pressure boundary condition. The corresponding density is calculated using the state law. To find the velocity components at the ghost cell center, the velocity at the reference point is transformed to get the tangential and normal velocities.

$$\boldsymbol{V}_{N,ref} = \left(\boldsymbol{V}_{ref} \cdot \boldsymbol{n}\right)\boldsymbol{n} \tag{3.3}$$

$$\boldsymbol{V}_{T,ref} = \boldsymbol{V}_{ref} - \boldsymbol{V}_{N,ref} \tag{3.4}$$

where the subscripts N and T represent normal and tangential components, respectively. $\boldsymbol{n}$ is normal vector of the surface panel with $\boldsymbol{n} = n_x\boldsymbol{i} + n_y\boldsymbol{j} + n_z\boldsymbol{k}$. Normal velocity on the wall should be zero in a non-permeable wall. Thus, linear extrapolation results in

$$\boldsymbol{V}_{N,g} = -\frac{\delta_g}{\delta_r}\boldsymbol{V}_{N,ref} \tag{3.5}$$

The slip-wall boundary condition is satisfied by setting tangential velocity as

$$\boldsymbol{V}_{T,g} = \boldsymbol{V}_{T,ref} \tag{3.6}$$

Laminar Wall

The pressure and adiabatic wall temperature boundary conditions of the laminar wall is identical to those of the inviscid wall. For an isothermal wall boundary, the temperature at the cell center is extrapolated from the reference point temperature and given wall temperature.

$$T_g = \left(T_w - T_{ref}\right)\frac{\delta_g}{\delta_r} + T_w \tag{3.7}$$

No slip condition for a laminar wall is enforced, such that the tangential velocity is

$$\boldsymbol{V}_{T,g} = -\frac{\delta_g}{\delta_r}\boldsymbol{V}_{T,ref} \tag{3.8}$$

The normal velocity follows the equation (3.5).

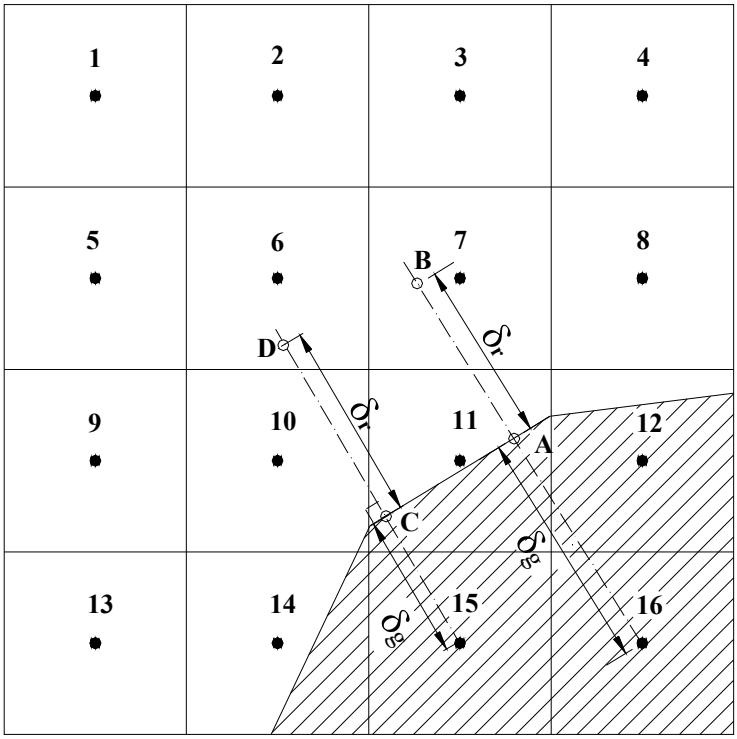


Figure 9 : Example configuration of wall boundary treatment

3.2 NEW TURBULENT WALL BOUNDARY CONDITION

3.2.1 THE LAW OF THE WALL

In high Reynolds number turbulent flows, it is assumed that the velocity profile near the wall obeys the law of the wall. In general, the region close to the solid wall is divided into two sub-layers in the wall function approach. They are a viscous sublayer, where purely viscous effects are dominant, and a turbulent sub-layer, where the log-law is applied.

$u^+ = y^+$ for laminar sub-layer $y^+ < y_c^+$

$$u^+ = \frac{1}{\kappa}\ln y^+ + B \text{ for turbulent sub-layer } y^+ > y_c^+ \tag{3.9}$$

where

$$y_c^+ = 11.63$$

$$u^+ = \frac{u_t}{u_\tau} \tag{3.10}$$

$$y^+ = \frac{\delta \rho_w u_\tau}{\mu_w} \tag{3.11}$$

δ represents the normal distance from solid wall, and the subscript "w" denotes the wall value. The friction velocity, u_τ, is defined as a function of wall skin friction, τ_w, and wall density, ρ_w,

$$u_\tau \equiv \sqrt{\frac{\tau_w}{\rho_w}} \tag{3.12}$$

$\kappa = 0.41$ is the Von Karman constant, and B is related to the roughness parameter known to be 5.0 for a smooth wall [95]. Many RANS solvers [15, 25, 33, 34, 41, 47, 52, 76] have applied the two layer formulation in which the buffer layer is not considered. Recently, Nichols and Nelson [62] employed the Spalding's formulation [77], which yields a unified form valid for the log lay layer and the viscous sublayer as well as the buffer layer. It is known that the Spalding's formulation shows excellent agreement with various experimental data even after, for $y^+ > 300$, the outer law commences [95].

$$y^+ = u^+ + e^{-\kappa B}\left[e^{\kappa u^+} - 1 - \kappa u^+ - \frac{\left(\kappa u^+\right)^2}{2} - \frac{\left(\kappa u^+\right)^3}{6}\right] \tag{3.13}$$

This formulation assumes incompressible adiabatic wall, but this can be extended to a compressible flow if the freestream Mach number is very large (i.e. $M_\infty \leq 5$). According to the Morkovin's hypothesis, the physics of compressible turbulence is very similar with incompressible flow if $(\gamma - 1)M_\infty^2$ is not large compared with unity [8]. The small effect of compressibility in non-hypersonic flow can be properly modeled by the density variation of friction velocity, u_τ [95]. The isothermal wall boundary in turbulent flow considering heat transfer effect is not considered, since it requires a large amount of work and validation to be another research area.

For the given temperature and tangential velocity at the reference point, the adiabatic wall temperature is calculated from the Crocco-Busemann equation.

$$T_w = T_{ref} + \frac{r}{2}\frac{V_{T,ref}^2}{c_p} \tag{3.14}$$

where r is the recovery factor known to be $\mathrm{Pr}^{1/3}$ in turbulent flows for air. c_p represents the specific heat. The wall density is obtained from the state law for calculated wall temperature and wall pressure that is equal to the reference point pressure. Then, the wall shear stress at the reference point can be calculated by solving equation (3.13) numerically, i.e. Newton's method.

3.2.2 APPLICATION OF WALL SHEAR STRESS

The computed wall shear stress is required to be enforced into the governing equation. In the implementation of wall function with coarse grid spacing near wall, incorrect velocity gradient and wall shear stress will be obtained when no-slip condition is applied. Figure 10

shows a typical example of turbulent velocity profile with coarse grid spacing near wall. The filled bulletins represent cell centers. At the point $1+\frac{1}{2}$, the shear stress τ_{xy} is approximated by

$$\tau_{xy,1+\frac{1}{2}} \approx \frac{1}{2}\left[\left(\mu_l + \mu_t\right)_1 + \left(\mu_l + \mu_t\right)_2\right] \frac{u_2 - u_1}{y_2 - y_1} \tag{3.15}$$

If the grid is not small enough to resolve viscous sublayer, the approximated slope of velocity profile is different from the actual velocity gradient on the wall. Hence, the shear stress will not be computed correctly.

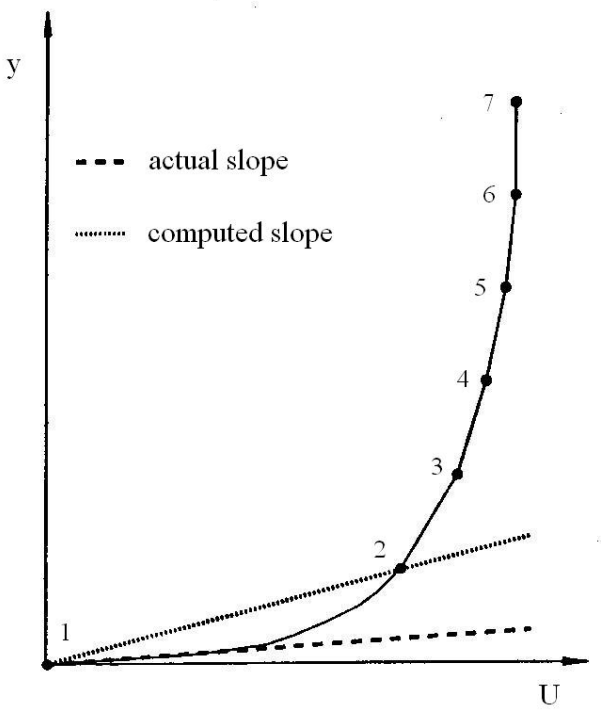


Figure 10 : Velocity profile with coarse wall spacing [76]

Previous Approaches

One approach to introducing the wall function corrected wall shear stress into the calculation of the viscous flux is to calculate the effective turbulent viscosity so that the discrete shear stress at the boundary cell face yields the correct value for the wall shear stress [15, 16]. Unfortunately, this method may results in errors into the energy equation if a separate effective turbulent viscosity for the temperature is not used.

Instead of modifying turbulent viscosity, calculated shear stress by wall function can be directly imposed on the face near wall. Sondak and Pletcher [76] introduced a procedure to perform a transformation of the stresses for generalized coordinate system with body-fitted structured grid. He used the standard tensor transformation to get

$$\tau_{\alpha\beta} = \frac{\partial \xi_\alpha}{\partial x_i} \frac{\partial \xi_\beta}{\partial x_j} \tau_{ij} \tag{3.16}$$

where α and β represent the geodesic coordinate, and i and j the Cartesian coordinate. In three-dimensional matrix form,

$$\begin{bmatrix} \tau_{\xi\xi} \\ \tau_{\xi\eta} \\ \tau_{\xi\zeta} \\ \tau_{\eta\eta} \\ \tau_{\eta\zeta} \\ \tau_{\zeta\zeta} \end{bmatrix} = \begin{bmatrix} \xi_x\xi_x & \xi_x\xi_y & \xi_x\xi_z & \xi_y\xi_y & \xi_y\xi_z & \xi_z\xi_z \\ \xi_x\eta_x & \xi_x\eta_y & \xi_x\eta_z & \xi_y\eta_y & \xi_y\eta_z & \xi_z\eta_z \\ \xi_x\zeta_x & \xi_x\zeta_y & \xi_x\zeta_z & \xi_y\zeta_y & \xi_y\zeta_z & \xi_z\zeta_z \\ \eta_x\eta_x & \eta_x\eta_y & \eta_x\eta_z & \eta_y\eta_y & \eta_y\eta_z & \eta_z\eta_z \\ \eta_x\zeta_x & \eta_x\zeta_y & \eta_x\zeta_z & \eta_y\zeta_y & \eta_y\zeta_z & \eta_z\zeta_z \\ \zeta_x\zeta_x & \zeta_x\zeta_y & \zeta_x\zeta_z & \zeta_y\zeta_y & \zeta_y\zeta_z & \zeta_z\zeta_z \end{bmatrix} \begin{bmatrix} \tau_{xx} \\ \tau_{xy} \\ \tau_{xz} \\ \tau_{yy} \\ \tau_{yz} \\ \tau_{zz} \end{bmatrix} \tag{3.17}$$

where ξ is a unit vector tangential to the wall, η normal to the wall, and ζ is the cross product of ξ and η. This also requires the inverse transformation from geodesic to Cartesian using a numerical method, i.e. Gauss-Jordan method.

As seen, the above method is very complicated and needs coordinate transformation, which would increase truncation error and emasculate the advantage of the Cartesian grid. Applying slip-wall boundary condition also requires the complicated formulations above. To make the matters worse, non-physical fluctuations of primitive variables are induced when Sondak's method is combined with the staggered Cartesian grid. As shown in Figure 7, the fluctuated pressure may result in fictitious separation and distort the whole solution. This is mainly caused by the opposite directional velocity of the cells whose center is located inside of the wall. The nonlinear velocity profile combined with sharp change in the distances from the cell center to the wall induces the non-physical fluctuations.

New Boundary Condition

To solve previously stated problems, a new wall function boundary condition is devised in the present study. It is neither no-slip wall nor slip wall boundary condition. It is based on the idea that the modified tangential velocity satisfying discrete wall shear stress approximation would eliminate the use of the complicated coordinate transformation. This makes the computational cells near wall to remain in numerically linear region, thereby, the computation would be stable. The approach is described in the following.

The normal velocity is specified by the equation (3.5) to ensure zero normal velocity on the wall. Since the total wall shear stress near the wall is approximately constant [82], the

wall shear stress at the ghost cell is set to be the same as that at the reference point. Assuming the total viscosity (the sum of molecular viscosity and eddy viscosity) of the ghost cell is identical to that of the reference point, the shear stress in the solver is approximated similar with the equation (3.15).

$$\tau_w \approx \left(\mu_{l,ref} + \mu_{t,ref}\right)\frac{\Delta V_T}{\Delta \eta} = \left(\mu_{l,ref} + \mu_{t,ref}\right)\frac{V_{T,ref} - V_{T,g}}{\delta_r - \delta_g} \tag{3.18}$$

Then, the tangential velocity of the ghost cell is

$$V_{T,g} \approx V_{T,ref} - \frac{\delta_r - \delta_g}{\mu_{l,ref} + \mu_{t,ref}}\tau_w \tag{3.19}$$

Applying linear tangential velocity at the ghost cells may result in non-physical mean velocity at boundary cells, which also occurs in the application of slip-wall condition to the turbulent wall boundary. However, this condition satisfies the required wall boundary conditions of zero normal flux and wall shear stress, and the flow cells outside of the boundary cells would have proper values. The pressure boundary condition is in the same manner with the inviscid and laminar wall conditions. The temperature of the ghost cell follows the Crocco-Busemann relation in (3.14) from the reference point temperature and computed tangential velocity. This is based on the fact that the adiabatic wall temperature is constant along the normal ray near wall.

$$T_g = T_{ref} + \frac{r}{2}\frac{V_{T,ref}^2 - V_{T,g}^2}{c_p} \tag{3.20}$$

The density is obtained from the state law.

3.2.3 SPECIFICATION OF TURBULENT PROPERTIES

The remaining step is to find the boundary conditions for turbulent properties, k and ε. The boundary conditions of k and ε have to satisfy the assumption of the equation (3.18), in which the sum of molecular viscosity and eddy viscosity is constant between the reference point and the ghost cell center. At the same time, the eddy viscosity should follow the designated profile, which is described in the reference [62].

$$\frac{\mu_t}{\mu_l} = \kappa e^{-\kappa B}\left[e^{\kappa u^+} - 1 - \kappa u^+ - \frac{\left(\kappa u^+\right)^2}{2}\right] \tag{3.21}$$

It is also known that the negative turbulent kinetic energy or dissipation causes non-physical solution and instability. To fulfill these requirements, the boundary conditions for turbulent properties are imposed on the flow cells contacting the boundary cells, as well as the ghost cells and boundary cells. This approach is similar with the structured grid wall function method, in which the turbulent properties are specified on the closest flow cell, not on the wall. Figure 11 shows the example of the flow cells (e.g. the cells from 7 to 10), on which the turbulent boundary conditions are enforced. After finding the closest point on the wall (point A) from the flow cell center (point 10), calculate the distance (δ_c) from the wall to the cell center. From the stored state vectors at the cell center, use the equation (3.4) and Crocco-Busemann equation to find the tangential velocity and wall density, respectively. In the equation (3.4), the reference point value is replaced by the cell center value. The solution of the Spalding's formulation gives the wall shear stress as well as the friction velocity. Assuming that the shear stress is constant near the wall, the turbulent kinetic energy at the cell center is equal to

$$k = \frac{u_\tau^2}{\sqrt{C_\mu}} \tag{3.22}$$

Given the eddy viscosity from the equation (3.21), the dissipation rate of the turbulence energy is set to satisfy the turbulent viscosity according to the Launder and Spalding's turbulence model.

$$\varepsilon = C_\mu \rho \frac{k^2}{\mu_t} \tag{3.23}$$

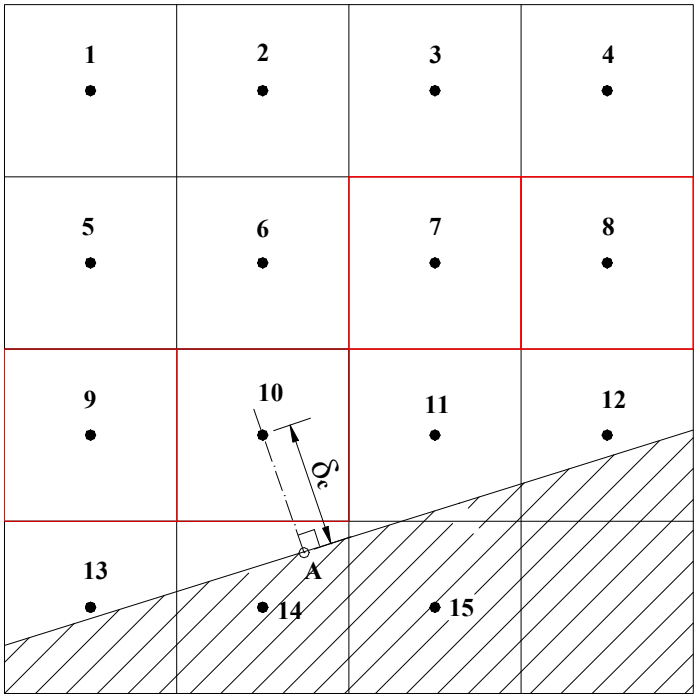


Figure 11 : Boundary conditions for turbulent properties on the flow cells

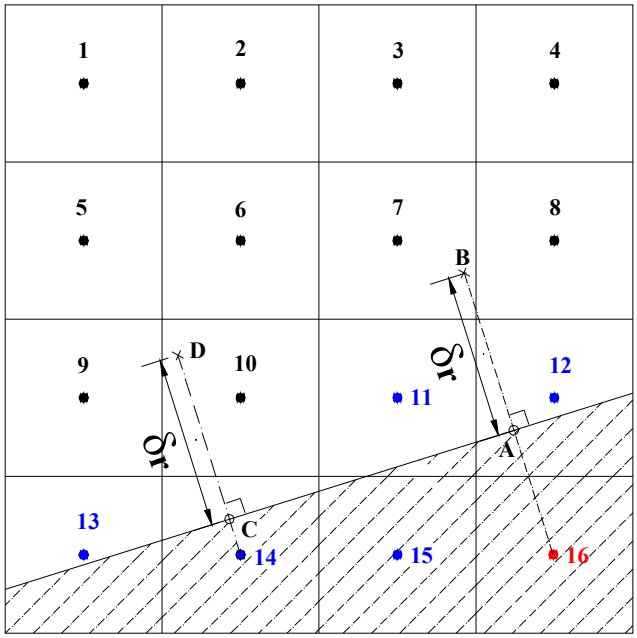


Figure 12 : Boundary conditions for turbulent properties on boundary cells and ghost cells

For boundary cells and ghost cells, the turbulent properties are specified using the reference point properties to ensure positivity and constant total viscosity along the normal rays. The boundary cells are the cells numbered from 11 to 15, and the ghost cell is the cell 16 in Figure 12. The point B and D denote the reference points of the ghost cell 16 and boundary cell 14, respectively. The calculation of the reference point values is described above. The constant total viscosity condition along the ray yields to

$$\mu_l + \mu_t = \mu_{l,ref} + \mu_{l,ref} \tag{3.24}$$

μ_l is obtained from the stored boundary cell temperature or ghost cell temperature from the equation (3.20) using the Sutherland's law. Then, the eddy viscosity of the boundary cell is expressed as

$$\mu_t = \mu_{l,ref} + \mu_{l,ref} - \mu_l \tag{3.25}$$

The turbulent kinetic energy and dissipation rate are specified using the equations (3.22) and (3.23), respectively.

3.2.4 SUMMARIZED PROCEDURE

The procedure of implementing turbulent wall boundary condition is summarized below.

1. Find the reference point of the ghost cell and calculate the density, velocity, pressure and temperature by the interpolation from its neighbor cells.
2. Use the equation (3.3) and (3.4) to get the normal and tangential velocity at the reference point. The adiabatic wall temperature and wall density are calculated from the equation (3.14).
3. Given tangential velocity, adiabatic wall temperature and density, solve the Spalding's formulation (3.13) to get the wall shear stress. The eddy viscosity of the reference point is obtained from the equation (3.21).
4. Apply the equation (3.5) and (3.19) to obtain the normal and tangential velocities of the ghost cell.
5. The pressure of the ghost cell is identical to that of the reference point. Temperature is specified from the equation (3.20), and density obeys the state law.

6. After finding the eddy viscosity of the ghost cell by the equation (3.25), use the equation (3.22) and (3.23) to get k and ε.
7. Apply wall boundary condition of turbulent properties on the flow cells, which is located next to the boundary cells. Note that the k and ε are calculated from the stored state vectors of the flow cells themselves, instead of the reference point values.
8. Repeat the step 1~3 for the reference point of the boundary cell, and apply wall function boundary condition of turbulent properties on the boundary cells, as explained at step 6.

3.3 INFLOW AND OUTFLOW BOUNDARY CONDITIONS

The inflow and outflow specified in NASCART is based on the characteristics of the flow. The flow information propagates along the characteristics, and the Riemann invariants remain constant along the characteristics. For 3-D Reynolds Averaged Navier-Stokes (RANS) equation using a $k-\varepsilon$ turbulence model, compatibility equations give seven Riemann invariants. Table 3 shows the Riemann invariants and corresponding characteristics.

For subsonic flow, the variables, R_1, R_2, R_3, R_4, R_6 and R_7 are propagated from the exterior and R_5 from the interior on the inflow boundaries. Similarly, R_1, R_2, R_3, R_4, R_6 and R_7 are propagated from the interior and R_5 from the exterior on the outflow boundaries.

All flow variables are, therefore, calculated from the specified Riemann invariants on the inflow and outflow boundaries.

Table 3 : Characteristics and Reimann invariants

Characteristics	Riemann invariants
$\left(\frac{dx}{dt}\right)_1 = u$	$R_1 = \frac{p}{\rho^\gamma}$
$\left(\frac{dx}{dt}\right)_2 = u$	$R_2 = v$
$\left(\frac{dx}{dt}\right)_3 = u$	$R_3 = w$
$\left(\frac{dx}{dt}\right)_4 = u + a$	$R_4 = u + \frac{2a}{\gamma - 1}$
$\left(\frac{dx}{dt}\right)_5 = u - a$	$R_5 = u - \frac{2a}{\gamma - 1}$
$\left(\frac{dx}{dt}\right)_6 = u$	$R_6 = k$
$\left(\frac{dx}{dt}\right)_7 = u$	$R_6 = \varepsilon$

For turbulent flow calculation, the turbulent kinetic energy and dissipation rate are assumed to be known at the inlet boundary from the prescribed turbulence intensity and the turbulent viscosity. A value for the ratio of freestream turbulent viscosity to laminar viscosity is also specified, such that

$$k_\infty = (I_t V_\infty)^2 \tag{3.26}$$

$$\mu_{t,\infty} = C_1 \mu_\infty \tag{3.27}$$

Applying turbulent viscosity closure in equation (3.23), the freestream turbulent dissipation is

$$\varepsilon_\infty = C_\mu \rho_\infty \frac{k_\infty^2}{C_1 \mu_\infty} \tag{3.28}$$

where I_t represents turbulent intensity and generally set to 0.01 for external flows [38]. Improper value of C_1 may result in slow convergence. In NASCART-GT, it is set to 0.1. Limiting the values of k and ε of the flow cell to the freestream values after each time step, helps prevent unphysical transient solution [76]. At the outflow boundary, k and ε are extrapolated according to the characteristic boundary condition.

C H A P T E R I V

ACTUATOR DISK MODEL

The rotor is modeled as an boundary type actuator disk with zero thickness, which represents an imaginary disk carrying pressure jump between the upper and lower surface allowing the flow to pass through. This approach reduces the computational resource requirement and eliminates the computational complexity of modeling each blade and performing time-accurate calculations. There are two methods in the developed code that decide the pressure jump across the actuator disk. One method is simply applying predetermined pressure jump at the position of given radius. It restricts the variation of azimuthal variation of thrust. The other method is to use the blade element theory. It calculates the thrust from the intermediate flow variables at every iteration. They will be described in the present chapter.

4.1 FUNDAMENTALS OF ACTUATOR DISK MODEL

The disk plane is represented by a finite number of rectangles, which exactly match one of the surfaces of hexahedral cells. The state vector at the cell center of a flow cell is calculated by integrating the fluxes on the six surfaces. Unlike the standard surface flux

calculation, the state vector on the disk surface is specified by special treatment suggested by Fejtek and Roberts [24].

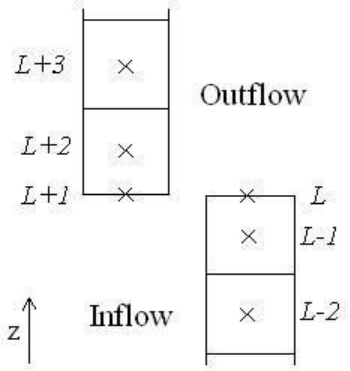


Figure 13 : Configuration of disk surface boundary condition

The points L and $L+1$ in Figure 13 denote the upper and lower surfaces of a blade respectively, and the main flow direction is aligned with negative z-axis. Consideration of the characteristics of the flow field indicates that, for a subsonic inflow boundary, six flow properties must be specified and one can be extrapolated from the interior solution domain. The rotor grid point of disk surface L is considered to be an inflow boundary, such that four flow properties (i.e. ρ , u , v , p , k and ε) are specified from the outflow boundary condition and one property (i.e. w) is extrapolated from the inflow domain.

$$\rho_L = \rho_{L+1}$$

$$u_L = u_{L+1}$$

$$v_L = v_{L+1}$$

$$(\rho w)_L = (\rho w)_{L-1} \tag{4.1}$$

$$p_L = p_{L+1} + \Delta p$$

$$k_L = k_{L+1}$$

$$\varepsilon_L = \varepsilon_{L+1}$$

The rotor grid point $L+1$ is regarded as an outflow boundary, where six flow properties are extrapolated and one (i.e. ρw) is fixed. The density ρ, the mass fluxes ρu and ρv, the total energy E_t, and mass fluxes of turbulent properties ρk and $\rho \varepsilon$ are updated using zero-order extrapolation from the solution domain at $L+2$. Mass continuity through the rotor disk is ensured by setting the z-directional mass flux to be conserved across the rotor disk.

$$\begin{aligned}
&\rho_{L+1} = \rho_{L+2} \\
&(\rho u)_{L+1} = (\rho u)_{L+2} \\
&(\rho v)_{L+1} = (\rho v)_{L+2} \\
&(\rho w)_{L+1} = (\rho w)_L \\
&(E_t)_{L+1} = (E_t)_{L+2} \\
&(\rho k)_{L+1} = (\rho k)_{L+2} \\
&(\rho \varepsilon)_{L+1} = (\rho \varepsilon)_{L+2}
\end{aligned} \tag{4.2}$$

In the current formulation, the tangential velocities normal to the rotor shaft are continuous across the rotor disk. Applying the torque boundary condition for the modeling of the rotor swirl described in the references[12, 24], gives the tangential velocity jump. In the present work, the torque boundary condition is not considered, since effect is known to be small for the cases studied and it may lead to nonphysical solution [13]. The small effect of

the torque boundary condition was shown by O'Brien [64], who computed GT rotor model using the actuator disk method. In the equations above, Δp represents the pressure jump across the rotor disk which generates the thrust of the rotor. The pressure jump is either specified as an input parameter or calculated using the blade element theory. These approaches are described in the next section.

4.2 SPECIFICATION OF PRESSURE JUMP

The summation of quantities multiplying pressure jump and cell surface area corresponding to the rotor disk, results in the total thrust of a given rotor. Total thrust must match the given thrust coefficient C_T. For a predetermined pressure jump case, the pressure jump is determined from the manipulation of given thrust coefficient and advance ratio of the rotor. Meanwhile, the pressure jump for the blade element method is calculated from the blade geometry and local flow properties using 2-D airfoil theory.

4.2.1 PREDETERMINED PRESSURE JUMP

To specify the pressure jump across the rotor disk, the simplest method is to assume uniform pressure jump that satisfies total thrust. This approach is commonly used for an axial fan modeling [42, 43]. However, the thrust distribution on the actuator disk is unlike the real helicopter rotor, since it neglects the variation of thrust in azimuthal angle and the existence of cut-off radius of the rotor blade. In present study, the uniform pressure jump

approximation is applied only for the validation of the actuator disk method excluding the fuselage effect.

The thrust coefficient is defined as

$$C_T = \frac{T}{\rho(\Omega R)^2 \pi R^2} \tag{4.3}$$

where T, Ω and R denote the total thrust, angular velocity and radius of the rotor, respectively. The uniform pressure jump, Δp, can be calculated as

$$\Delta p = \frac{T}{\pi R^2} \tag{4.4}$$

Thus, the thrust coefficient is expressed as a function of pressure jump.

$$C_T = \frac{\Delta p}{\rho(\Omega R)^2} \tag{4.5}$$

Applying advance ratio, $\mu \equiv V_\infty / \Omega R$, the resultant pressure jump is shown as

$$\Delta p = C_T \frac{\rho V_\infty^2}{\mu^2} \tag{4.6}$$

4.2.2 BLADE ELEMENT THEORY (BET)

Pressure Jump Calculation

Unlike the predetermined pressure jump, the blade element theory considers the variation of thrust in azimuth angle and cut-off radius. This approach is based on the 2-D airfoil theory, such that it neglects the 3-D wing tip effect of a blade. Even though the calculated pressure jump is nothing to do with the thrust coefficient at an intermediate stage,

the final results should be converged to the given value. Configuration of a rotor blade is presented in Figure 14.

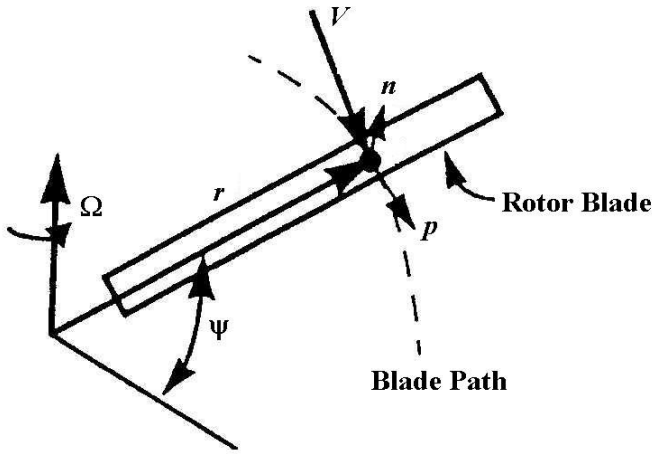


Figure 14 : Coordinate system of rotor blade from J. Lee and O.J. Kwon [50]

The following description is from J. Lee and O.J. Kwon [50], and summarized for reader's convenience. The local velocity, $\boldsymbol{V}$, is continuous through the rotor disk plane, and its magnitude and direction are known as a part of the computation. Then, the induced angle of attack, α_i, at the center of each rectangular surface on the rotor disk, can be calculated as

$$\alpha_i = Tan^{-1}\left(\frac{\boldsymbol{V}\cdot\boldsymbol{n}}{\boldsymbol{V}\cdot\boldsymbol{t}}\right) \tag{4.7}$$

As shown in Figure 14, the vector $\boldsymbol{n}$ denotes the unit vector normal to the rotor disk plane, and $\boldsymbol{t}$ unit vector tangential to the rotor disk plane. The tangential component of velocity vector includes contributions from the local flow velocity and the rotor rotational speed

$$\boldsymbol{V}\cdot\boldsymbol{t} = \boldsymbol{V}\cdot\boldsymbol{p} + \left(\boldsymbol{r}\times\boldsymbol{\Omega}\right)\cdot\boldsymbol{p} \tag{4.8}$$

where $\boldsymbol{r}$ is the distance from the rotor axis of rotation to a point of a blade, and $\boldsymbol{p}$ the unit vector parallel to the blade path. The effective angle of attack, α_e, is then

$$\alpha_e = \Theta - \alpha_i \tag{4.9}$$

In general, the pitch angle of a blade, Θ, is generally expressed as

$$\Theta = A_0 - A_1 \cos\psi - B_1 \sin\psi + \left(0.75 - \frac{r}{R}\right)\alpha_t \tag{4.10}$$

where A_0 is the blade collective pitch setting. A_1 and B_1 are the coefficient of lateral and longitudinal cyclic pitch angles. ψ and α_t denote the blade azimuth angle and twist angle at the tip, respectively.

Since the velocity relative to the blade is known, elemental lift and drag acting at each section of the rotor blade can be calculated as

$$\Delta L = \frac{1}{2}\rho V_{rel}^2 C_l \Delta A \tag{4.11}$$

$$\Delta D = \frac{1}{2}\rho V_{rel}^2 C_d \Delta A \tag{4.12}$$

where C_l and C_d represent the sectional lift and drag coefficients for a given airfoil. ΔA is the elemental area of each rectangle on the rotor disk. The velocity shown above is the relative velocity expressed as

$$\boldsymbol{V}_{rel} = \boldsymbol{V} + (\boldsymbol{r} \times \boldsymbol{\Omega}) \cdot \boldsymbol{p} \tag{4.13}$$

The lift and drag coefficients are determined by using the 2-D airfoil theory. For N blades of the rotor, the elemental thrust, ΔT, for each rectangle is scaled by a time factor,

$N(d\psi / 2\pi)$, to obtain time-averaged contribution while the rotor sweeps the azimuth angle of $d\psi$.

$$\Delta T = N \frac{d\psi}{2\pi}\left(\Delta L \cos\alpha_i - \Delta D \sin\alpha_i\right) \tag{4.14}$$

The time-averaged contribution above assumes that time for a blade sweeping a surface center is small relative to time for a round of a blade, i.e.

$$\frac{c}{r\Omega} << \frac{2\pi R}{R\Omega} \tag{4.15}$$

This assumption limits the ratio of chord length, c, to the radius of rotor disk surface, r.

$$\frac{c}{r} << 2\pi \tag{4.16}$$

Now, the azimuthal sweeping angle $d\psi$ can be replaced by the chord length and the radius of the rotor disk.

$$r\, d\psi \approx c \tag{4.17}$$

Substituting equation (4.17) into (4.14) gives the thrust expression for the present unstructured quadrangular surface mesh.

$$\Delta T = \frac{N\Delta A}{4\pi r} c\rho\, V_{rel}^2 \left(C_l \cos\alpha_i - C_d \sin\alpha_i\right) \tag{4.18}$$

The Prandtl-Glauert rule [4] is applied to include compressibility effect on the sectional lift and drag coefficient. Note that the thrust expression in equation (4.18) has a singular point at the rotor center. As the radius approaches to the center (i.e. $r \to 0$), the assumption in equation (4.16) does not hold anymore. The ratio of chord length to radius is, therefore, limited to 1/2.

$$\frac{c}{r} = \min\left(\frac{c}{r}\ ,\ 0.5\right) \tag{4.19}$$

The difference in pressure between the upper and lower surface meshes of the rotor disk plane can be represented as

$$\Delta p = \frac{\Delta T}{\Delta A} \tag{4.20}$$

Rotor Trim

Rotor trim is added in the blade element method, and corrects collective and cyclic pitch angles in order to obtain the desired thrust and eliminate moments about the hub. Using the pitch angles of the experimental model or real rotor in the blade element method, generally, induces errors in the thrust and moments of the rotor. This is mainly due to the limitations of the actuator disk model. Since the time-averaged formulation and simplified lift and drag distributions of the blade element method cannot properly simulate the unsteady flow over individual blade, the predicted thrust and moments are different from the measured values. The error is larger in the compressible flow solver than incompressible solver. In most compressible flow solver, the freestream Mach number is increased to avoid incompressible limit. The flow conditions around the actuator disk would be different due to the compressibility effects, although other nondimensional parameters are identical to the experiments. Therefore, the rotor trim routine is strongly required to properly compute the rotor-fuselage interaction.

The first step is to calculate the thrust coefficient and lateral and longitudinal moment coefficients about the hub are obtained by integrating the pressure jump on the actuator disk.

Assuming simple linear relationships between the angles and coefficients, the new angles are found to get the desired thrust and zero moments. Then, calculate the coefficients again using the modified angles. This procedure is repeated until the angles are converged or a specified iteration number has been met [13].

4.3 VALIDATION OF ACTUATOR DISK

For the validation of the actuator disk model, Euler computation is performed on isolated rotor test cases in axial flight. The freestream Mach number is set to 0.1. Imposing a constant pressure jump of $\Delta p = 300\ N/m^2$ corresponds to the rotor thrust ($C_T = 0.006$) presented in reference [7]. The computational boundaries are 5 diameter of disk apart from actuator disk center in x, y and z directions. The cell numbers in radial direction change from 16 to 128 to test grid dependency. The order of accuracy of flux calculation is tested for 1st order and 2nd order fully upwind schemes. This cell number limits the solution adapted refined cell size. Figure 15 shows the final computational grid after solution adaptations based on divergence. Grids are refined near the actuator disk and downstream of it. The predicted normal velocity distribution is presented in Figure 16. It is noted that the flow is gradually accelerated across the actuator disk.

The computed normalized velocity, V_{norm}, on the axis ($r/R = 0$) is compared with ONERA case in reference [7]. The normalized velocity is defined as

$$V_{norm} = \frac{1}{2}\frac{V_{computed} - V_{\infty}}{V_F} + 1 \tag{4.21}$$

where V_F is the Froude velocity,

$$V_F = \frac{1}{2}\left(\sqrt{V_\infty^2 + \frac{2\Delta p}{\rho}} - V_\infty\right) \tag{4.22}$$

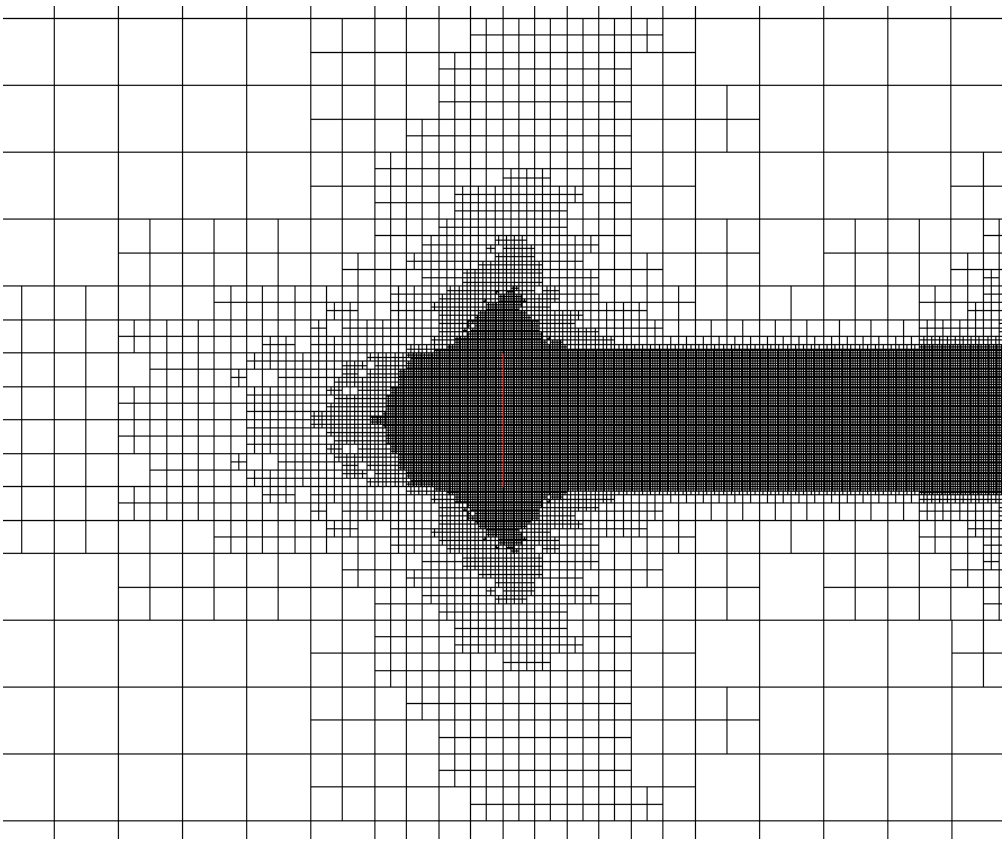


Figure 15 : Final computational grid for actuator disk validation

In Figure 17, N represents the cell number on the disk in radial direction. The actuator disk is located at $x/R = 0$, and negative region is upstream and positive downstream. The computation results approaches to ONERA case as the cell number increases. Using higher order scheme gives better result for the same grid resolution. The computed normal velocities of 1st order finest grid and 2nd order upwind case show good agreement with ONERA.

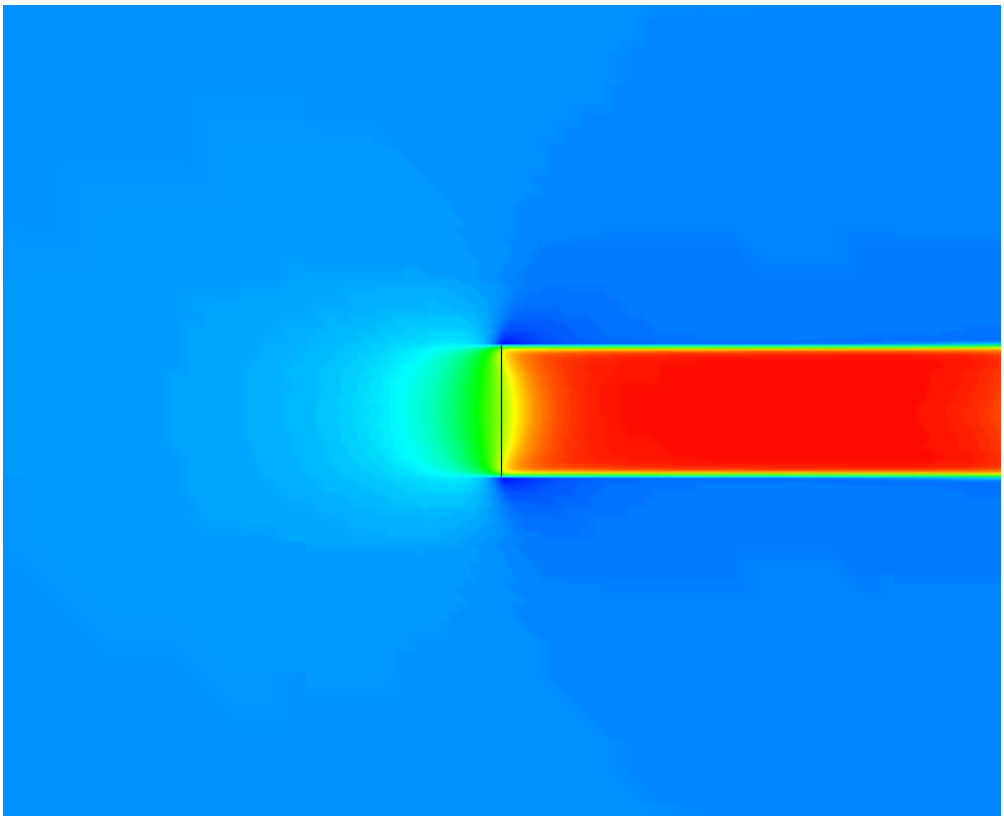


Figure 16 : Contour of normal velocity around actuator disk

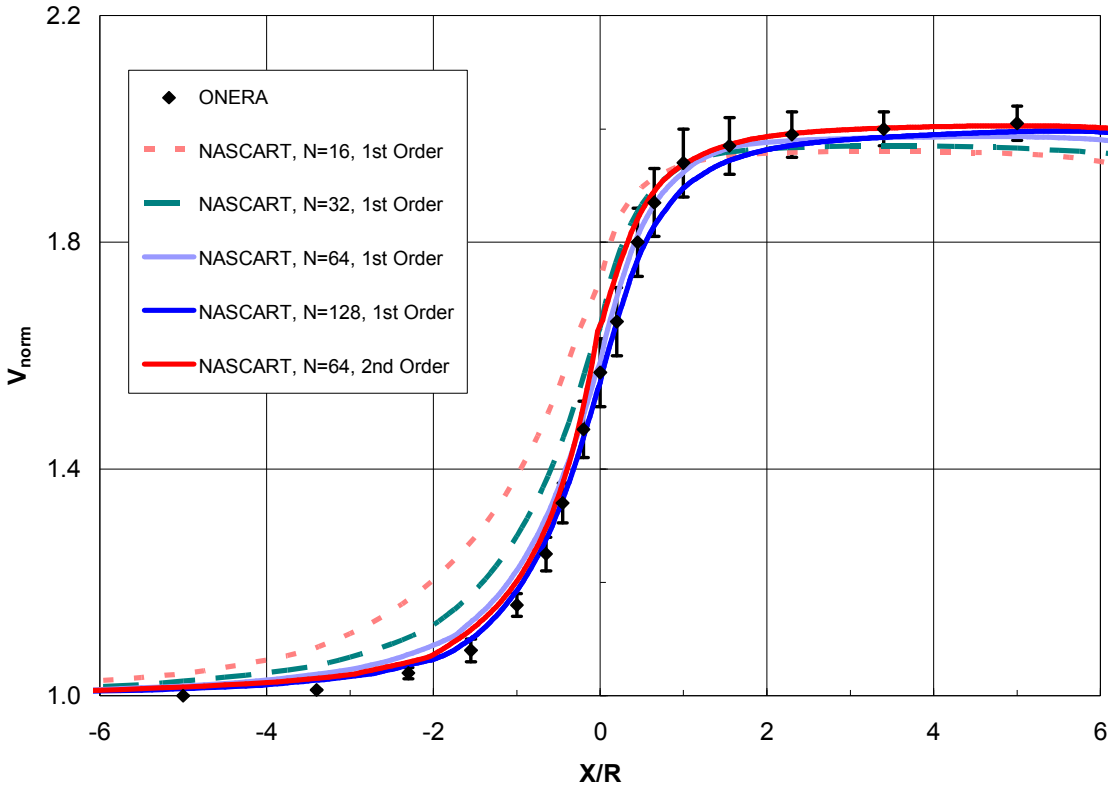


Figure 17 : Computation results of actuator disk validation

C H A P T E R V

RESULTS

The validation of the developed numerical approach in NASCART-GT is performed on the various 2-D viscous flows and 3-D viscous flows over two rotorcraft models with actuator disk. 2-D validations include the laminar and turbulent flows over flat plate and NACA 0012 airfoil, and the axisymmetric turbulent flow over a hemispheroid. The results are compared with analytical solution or experimental data. The actuator disk modeling is validated using the developed RANS solver on the GT rotor model and ROBIN rotor model. By comparing the results with the Euler solution, the effect of viscous phenomena on the rotor-fuselage interaction is studied. All the results shown in this chapter are obtained using the 3rd order interpolation of the inviscid fluxes.

5.1 2-D VALIDATION

The two-dimensional validations of viscous flow calculation are performed in both laminar and turbulent flow. The objective of the laminar calculations is to validate the ghost wall boundary condition in the immersed Cartesian grid system. For the turbulent calculation, the implementation of the developed wall function boundary condition is tested by comparing

with an analytical solution and the experimental data. The grid efficiency of the developed wall function boundary condition is described at the end of this section.

5.1.1 FLAT PLATE

A body-aligned Cartesian grid is generated to calculate turbulent flow over a two dimensional flat plate. The computational domain is shown in Figure 18. Inflow boundary is located ahead of the leading edge, and outflow boundary is on the trailing edge. On the inflow boundary, the characteristic boundary condition is enforced as usual. On the upper and outflow boundaries, however, the primitive variables are extrapolated instead of characteristic boundary condition. This ensures the flow smoothly sweep out. In front of the leading edge, the symmetric boundary condition of zero-gradient is used. In both laminar and turbulent calculations, the freestream Mach number is set to 0.2 to avoid incompressible limit of the solver.

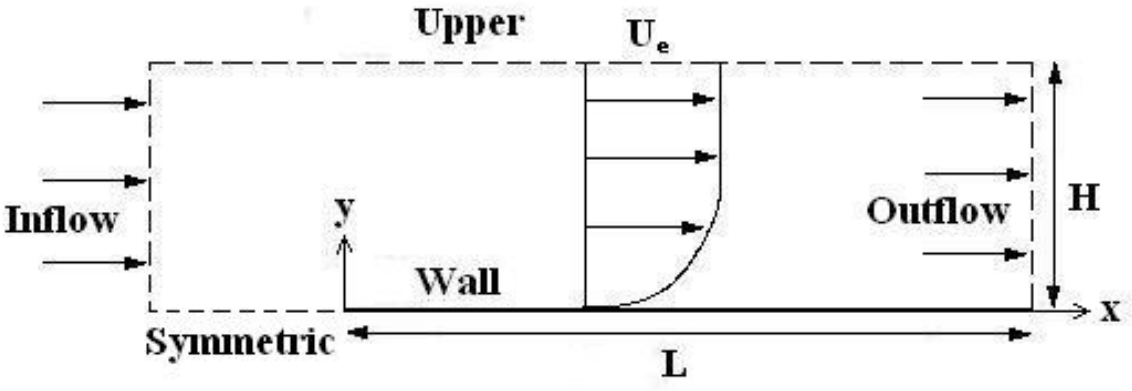


Figure 18 : Sketch of the flow field and computational domain

Laminar Flow

The Reynolds number is set to 20,000 based on the chord length, which is $L = 1\,m$. The height of the computational domain is $H = 0.5\,m$. The vertical computational boundaries are the trailing edge and one chord ahead of the leading edge, in which the square root cells of 16×4 are generated and 8 level of grid refinement is performed. Figure 19 shows the generated grid over flat plate, in which the red line represents the wall boundary. The flow cells are refined within the predetermined distance from the wall to resolve the boundary layer. In the current calculation, the distance is set to be the boundary layer thickness from the analytical Blasius solution for given Reynolds number.

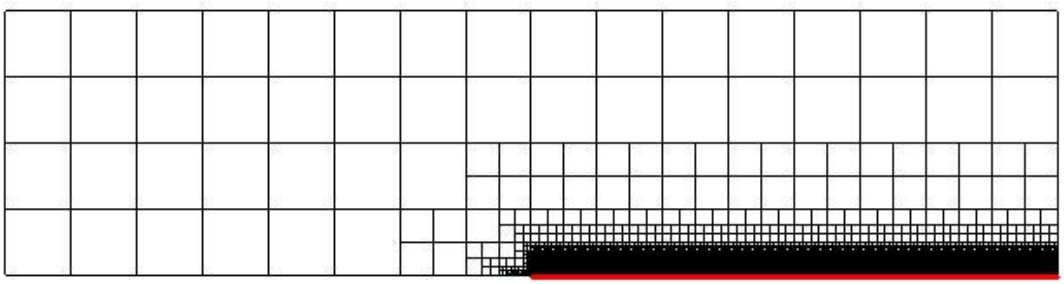


Figure 19 : Generated grid over flat plate

In Figure 20 and Figure 21, the computed skin friction coefficient and the velocity profile at half-chord are presented. They show good agreement with analytical Blasius solution. The computed boundary layer thickness is also compared very well with the analytical solution. The Blasius solution of skin friction is expressed as

$$C_f = \frac{0.664}{\sqrt{\mathrm{Re}_x}} \tag{5.1}$$

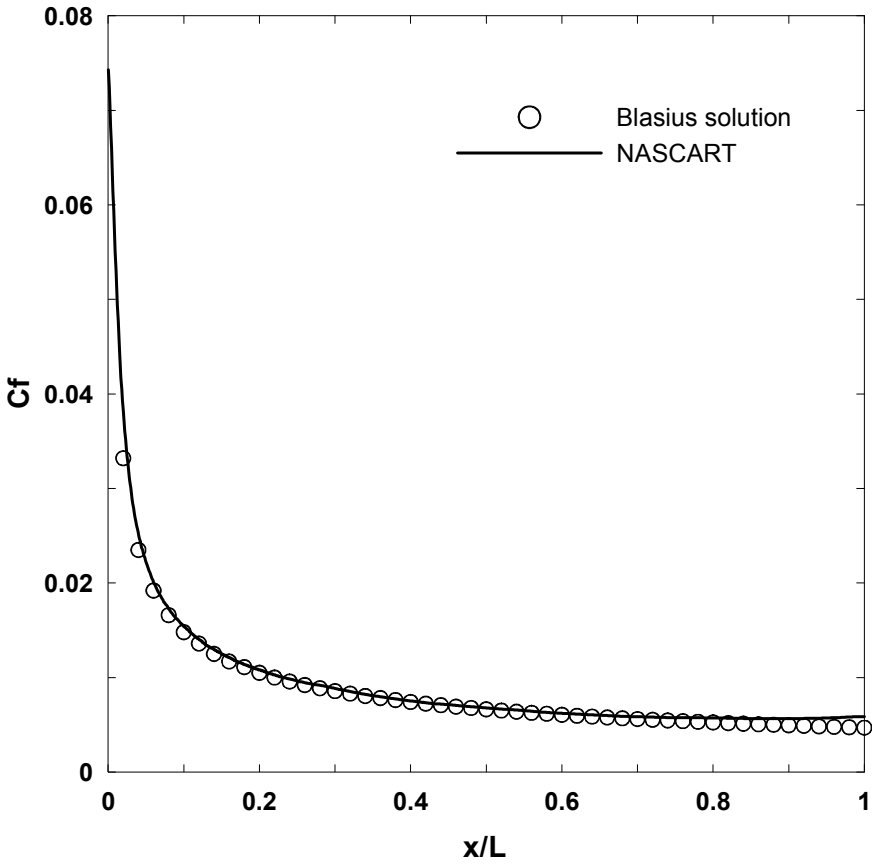


Figure 20 : Computed skin friction coefficient over laminar flat plate

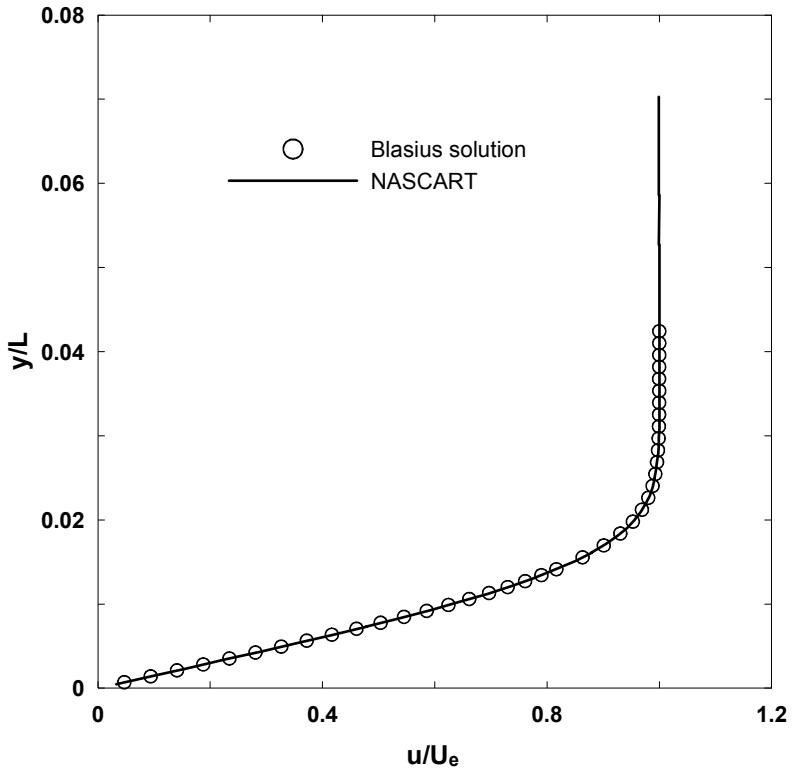


Figure 21 : Laminar velocity profile at half chord

Turbulent Flow

The Reynolds number based on flat plate length is 1.0927×10^7, at which the skin friction and velocity profiles have been measured by Wieghardt [97]. The computational domain and root cell dimension are identical to the laminar calculation. However, the grid is finer than the laminar case such that the maximum grid refinement level is 9. This results in the largest y^+ of 210.1 based on the reference point, which is larger than the cell size requirement of most RANS solvers. Since the wind tunnel turbulence level was not measured in the experiment, a value of $I_t = 0.01$ was assumed to specify freestream turbulent kinetic

energy, similar to general external flow calculation. A nondimensional freestream turbulent viscosity of 0.1 was assumed, and ε_∞ was calculated from equation (3.28). The surface roughness parameter was not measured, either. A value of $B = 5.0$ was used in the law of the wall assuming smooth wall.

The computed skin friction is compared with the power law, exact law by White [95], and the experiments in Figure 22. The equation of the power law is

$$c_f = \frac{0.058}{\mathrm{Re}_x^{1/5}} \tag{5.2}$$

Exact law by White is expressed as

$$c_f = \frac{0.455}{\ln^2(0.06\,\mathrm{Re}_x)} \tag{5.3}$$

Near the leading edge, the power law is more consistent with the experiment than the exact law by White. Aft of the mid-chord, the measured skin friction is between the power law and the exact law, which is well analyzed by NASCART. As shown in the picture, the skin friction changes rapidly near the leading edge. The inaccuracy of the computed skin friction near the leading edge might be caused by the insufficient local grid resolution. This will be discussed later.

Figure 23 shows the comparison of computed and measured mean velocity profile at various locations. The locations of the measured stations are shown in Table 4. It is observed that the calculated mean velocity profile and boundary layer thickness have good correlation with the measurements. Slight over-prediction of the boundary layer thickness from station 1 thru 4 is also induced by the insufficiently large cells, which smears out the high velocity gradient.

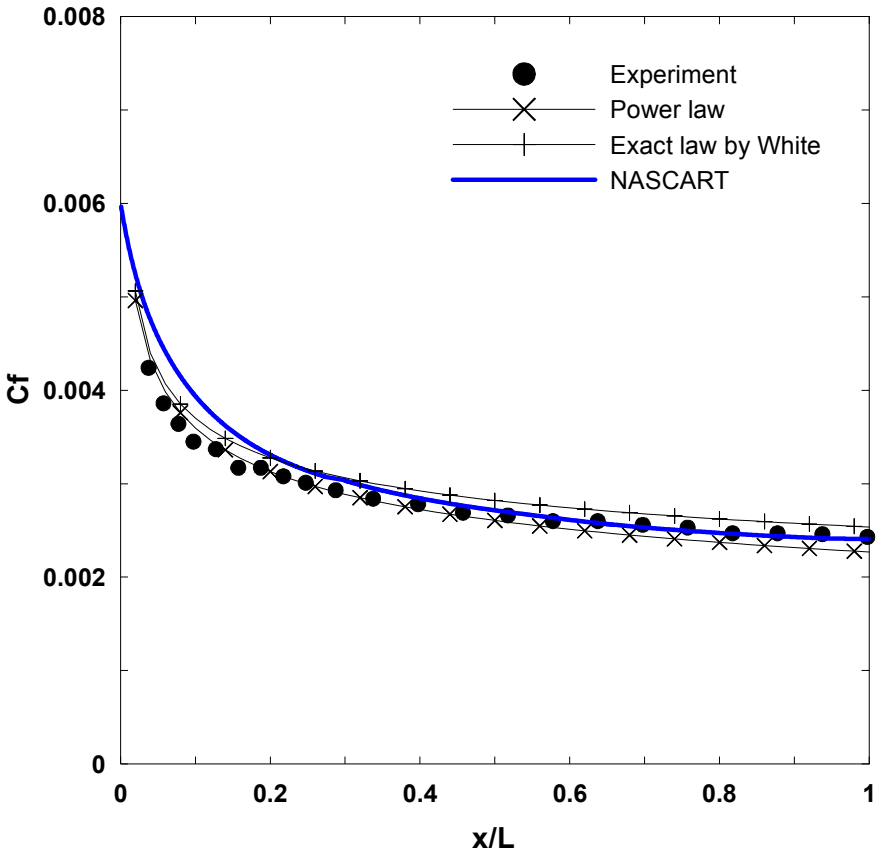
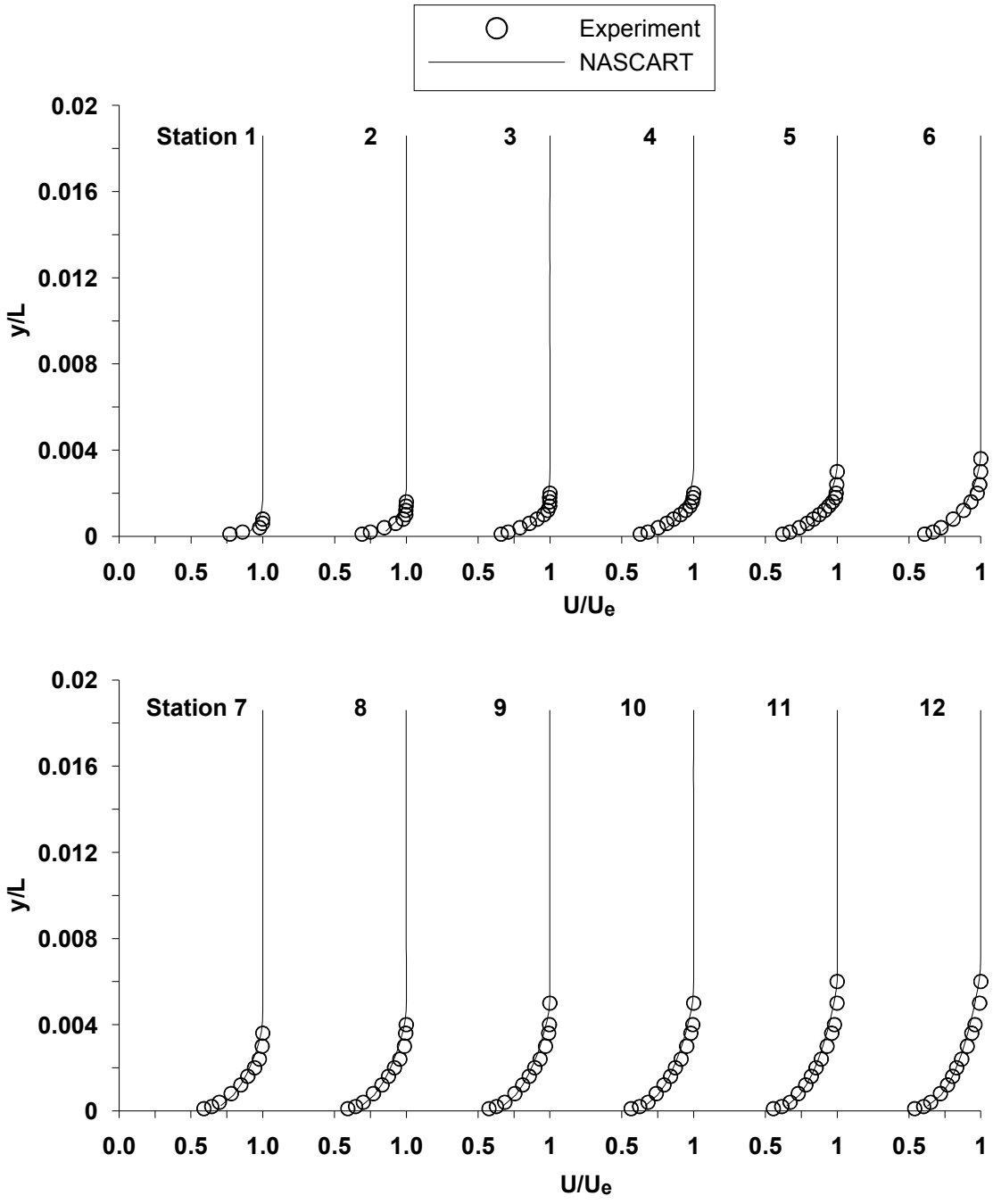


Figure 22 : Computed plate skin friction coefficient over turbulent flat plate

Table 4 : Locations of velocity profile measurement

Stations	x/L	Stations	x/L	Stations	x/L
1	0.0174	9	0.2174	17	0.6374
2	0.0374	10	0.2474	18	0.6974
3	0.0574	11	0.2874	19	0.7574
4	0.0774	12	0.3374	20	0.8174
5	0.0974	13	0.3974	21	0.8774
6	0.1274	14	0.4574	22	0.9374
7	0.1574	15	0.5174	23	0.9974
8	0.1874	16	0.5774		


Experiment
NASCART
Station 1
2
3
4
5
6
Station 7
8
9
10
11
12
y/L
0.02
0.016
0.012
0.008
0.004
0
0.0
0.5
1.0
1
U/Ue

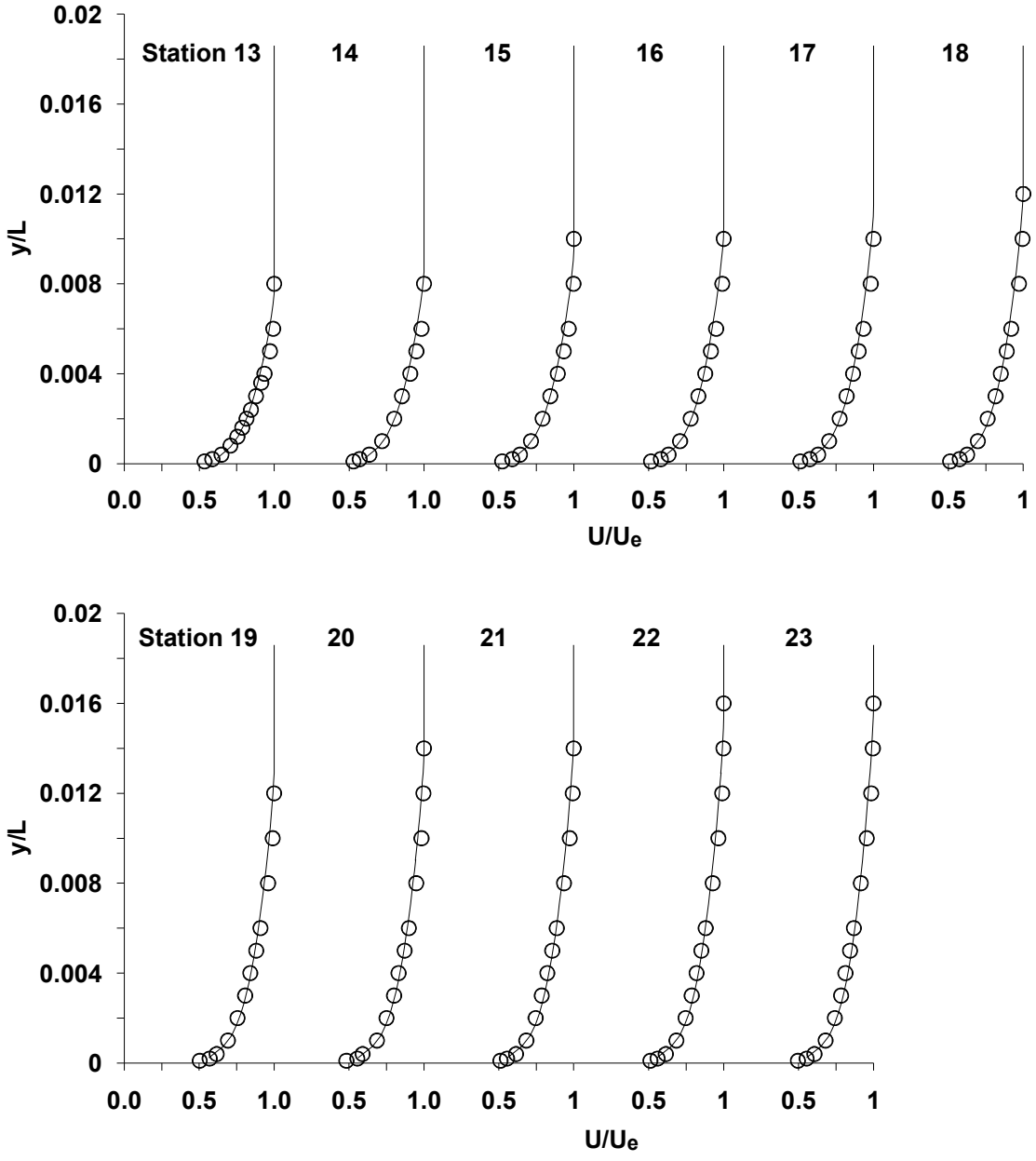


Figure 23 : Comparison of mean velocity profiles on turbulent flat plate

5.1.2 NACA 0012 AIRFOIL

Laminar Flow

The freestream Mach number is 0.8 and the angle of attack is 10 degree. Reynolds number based on chord length is 500. Since there is no experimental data available in such a low Reynolds number, the computed results are compared with the solution of verified multi-grid solver by Casalini [11]. D. Marshall [55] also performed the former version of NASCART calculation on the same flow condition. As explained in the introduction, Marshall employed the embedded Cartesian grid method in which the primitive variables at the cut-cell centroid are extrapolated from the reference point instead of using ghost cell.

The computational boundaries are 5 chords ahead of and behind the airfoil, above and below the airfoil centerline. The root grid dimension is 33×30, and 6 level of refinement is applied for the coarse grid. The computational cells of the coarse grid calculation are identical to those of Marshall's calculation. For the fine grid case, 8 level of refinement is employed with the root grid dimension of 22×20. The solution adaption is performed based on divergence and vorticity. Figure 24 shows the final grid configuration and entropy contours after solution adaption. The grids are refined near the stagnation point and in the shock pocket due to the large divergence variation. The cells are also refined inside the boundary layer and wake region where the high vorticity appears.

The computed distributions of Cp and Cf with coarse grid are presented in Figure 25 and Figure 26. In Figure 26, negative skin friction does not mean a separated flow, but is instead used to denote skin friction in the lower surface of the airfoil. The solution of current NASCART has better accuracy than Marshall's results in the prediction of the stagnation

pressure and suction peak in Figure 25. The computed skin friction also shows better agreement with the reference data. With respect to the unphysical fluctuations, the present results give smoother variation than the Marshall's in both of pressure and skin friction. The improvement is the result of the use of the conservative ghost cell approach and coordinate-aligned cell centers in the new version of NASCART. As described before, the present solver performs the volume integration for the boundary cells to specify the properties at the center of the non-cut boundary cells, while Marshall's approach extrapolates the flow properties at the centroids of the cut-cells from neighbor flow cells. The under-estimations of suction peak pressure and skin friction near the leading edge can be solved by increasing grid resolution as shown in Figure 27 and Figure 28. The computed Cp's and Cf's are compared well with the reference data.

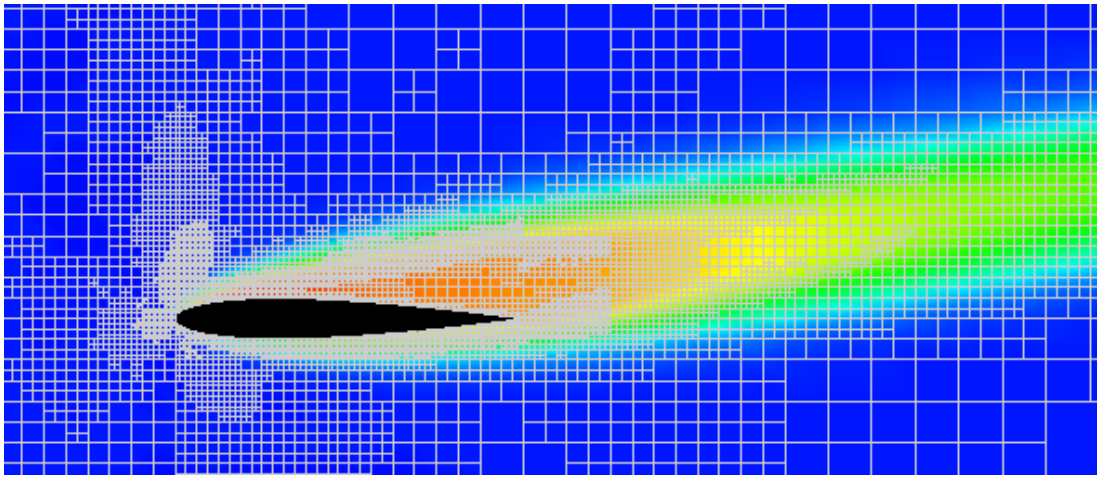


Figure 24 : Final grid configuration and entropy contours around transonic NACA 0012 airfoil

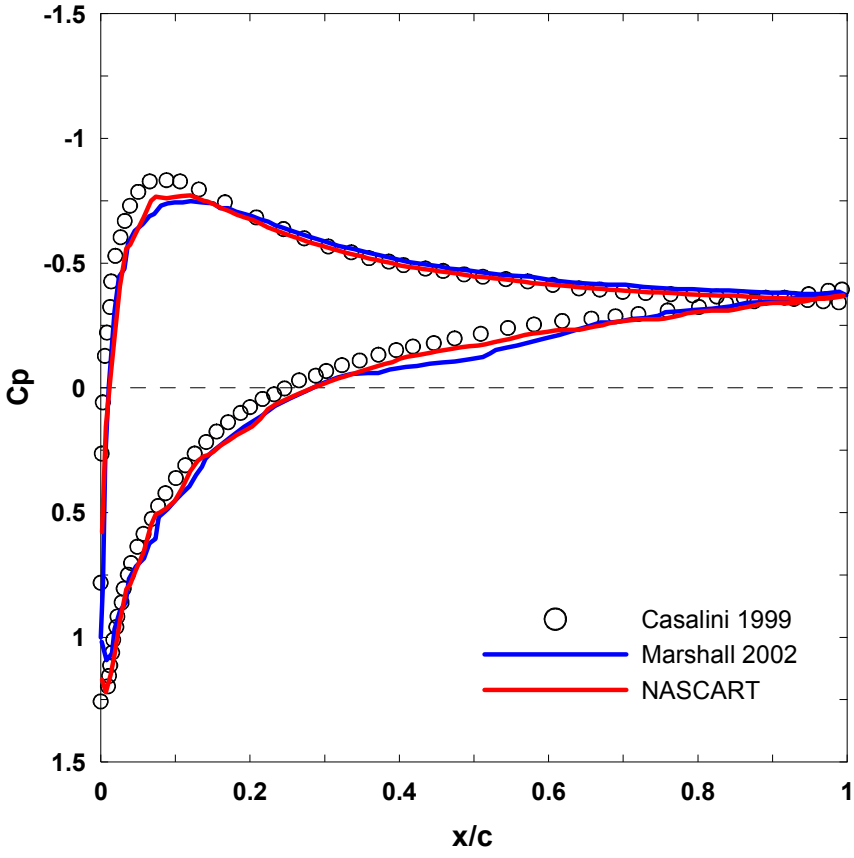


Figure 25 : Pressure distribution over transonic NACA 0012 airfoil with coarse grid

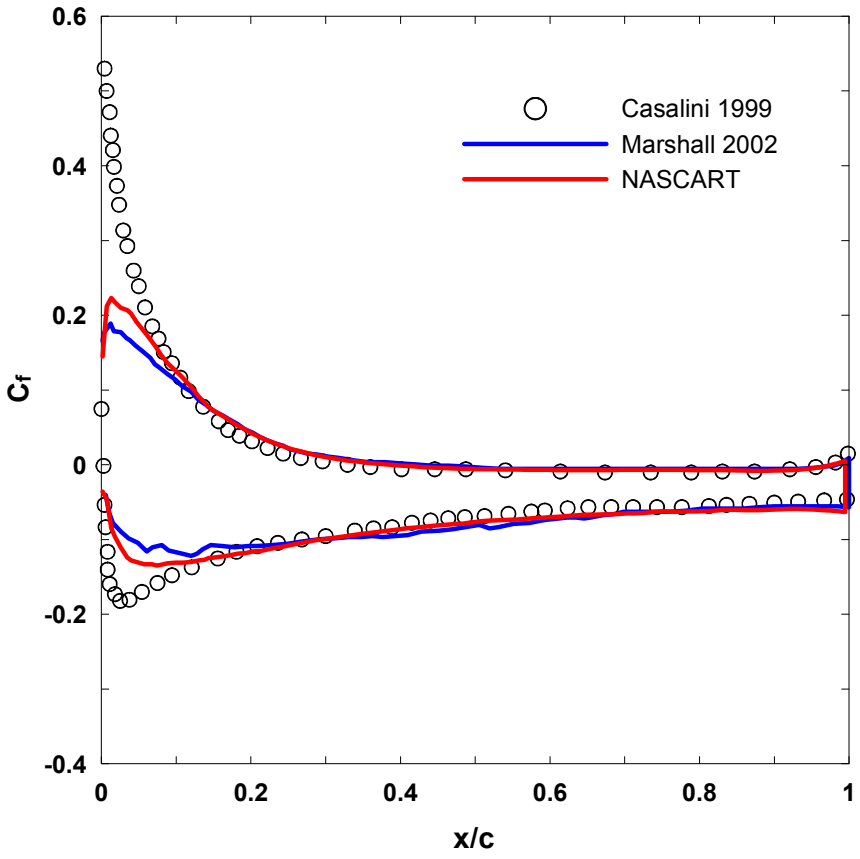


Figure 26 : Skin friction distribution over transonic NACA 0012 airfoil with coarse grid

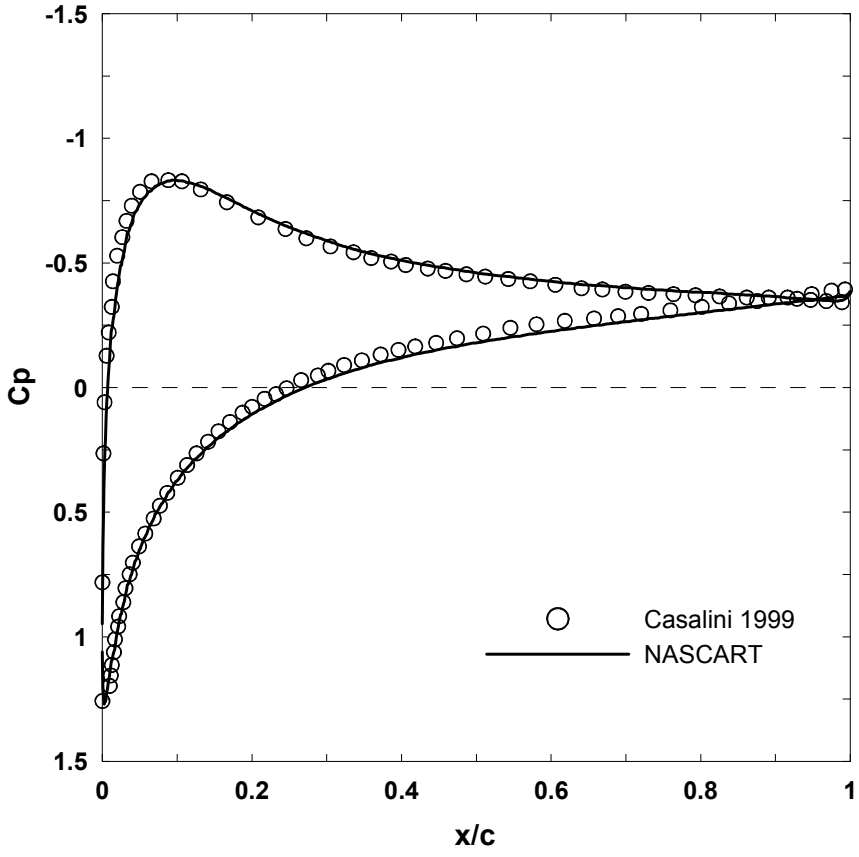


Figure 27 : Pressure distribution over transonic NACA 0012 airfoil with fine grid

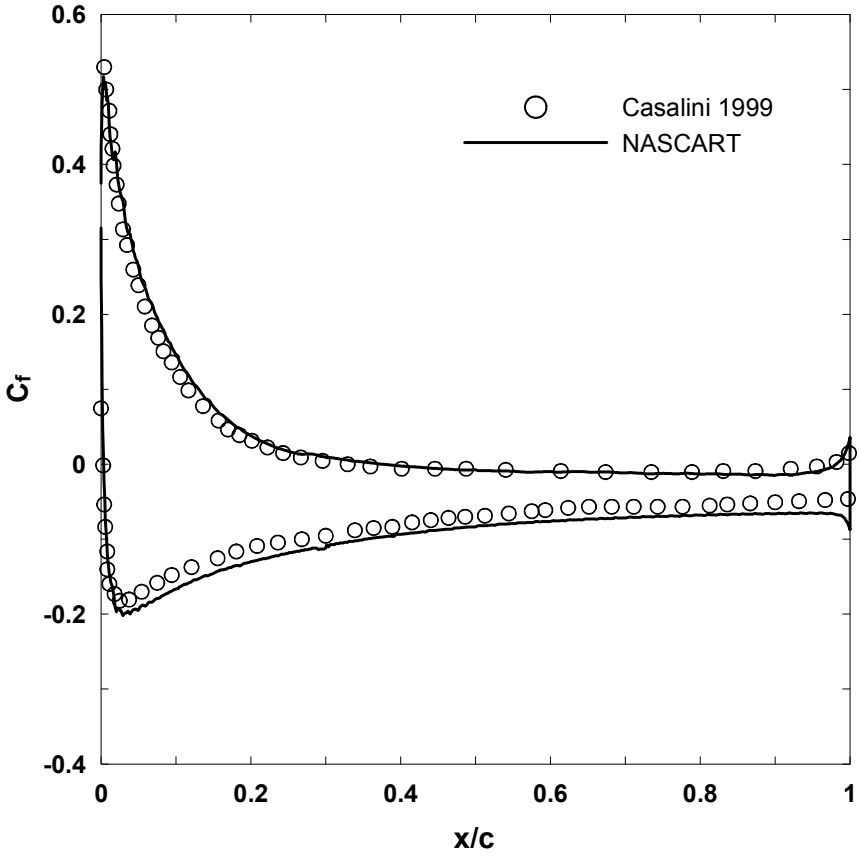


Figure 28 : Skin friction distribution over transonic NACA 0012 airfoil with fine grid

Turbulent Flow

Turbulent modeling is tested over a NACA0012 airfoil at a Mach number of 0.3, a chord Reynolds number of 1.86 million, and an incidence of 3.59 degrees. The computational boundaries are the same as the laminar calculation. The root grid dimension is 22×20, and 10 level of refinement is applied to get the maximum y^+ based on the reference point of 81.1. The characteristic boundary conditions are applied on the inflow and outflow

boundaries. The freestream turbulent intensity of 0.01 is assumed as usual in external flow calculation, and the freestream turbulent viscosity is set to 10% of the freestream molecular viscosity. A smooth wall is assumed to specify $B = 5.0$ for the law of the wall.

The computed pressure coefficients are compared with AGARD experimental data [1] in Figure 29, and very good correlation is observed.

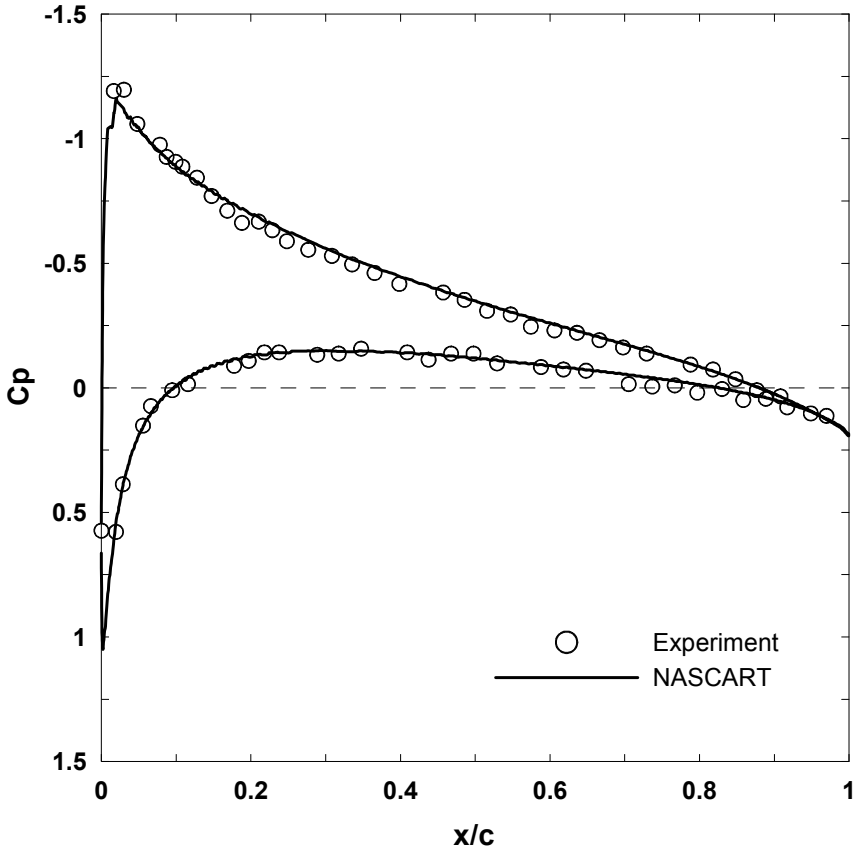


Figure 29 : Comparison of computed Cp with experiment for turbulent flow over a NACA 0012 airfoil

5.1.3 HEMISPHEROID

The developed turbulent methodology is tested over the hemispheroid to validate the capability of blunt body calculation. The turbulent boundary layer on the hemispheroid was measured by Ramaprian et al. [70] to provide validation data for computational development. The experimental data correspond to axisymmetric flow as well as the flow at an angle of incidence of 15°, and include surface pressure distribution, skin friction, and the distributions of the magnitude and orientation of the velocity vector in boundary layer. The results exhibit most of the complexities encountered in practical boundary layer flows including separation, and longitudinal vortices.

Figure 30 shows the model configuration, which combines a hemispherical nose with a hemispheroidal rear. The experiments are performed at the Reynolds number (based on the length of the body) of 2.0×10^{6}. The nominal wind-tunnel velocity is 22.0 $m/\sec$ thereby $M_{\infty}\approx 0.063$. The computational Mach number is increased to avoid the incompressible limit of the code, and the computations are performed for axisymmetric flow. The velocity profile and skin friction are measured at 8 traverse stations, which are shown in Table 5.

The computational boundaries are 5 times body length ahead of and behind the hemispheroid, above and below the centerline. The root grid dimension is 28×24, and 11 level of refinement is applied. The freestream Mach number is increased to 0.3 to prevent poor conditioning of the compressible flow solver. The freestream turbulent intensity is 0.01, and the freestream turbulence dissipation rate is determined for the eddy viscosity to be 10% of the molecular viscosity. The constant related to the roughness parameter is set to $B=5.5$ according to the experiments. The largest y^{+} based on the reference is 42.8.

Figure 31 shows the final grid configuration, in which the red line represents the wall boundary. The cells are refined near the leading edge and wake region where the large gradients of divergence and vorticity exist.

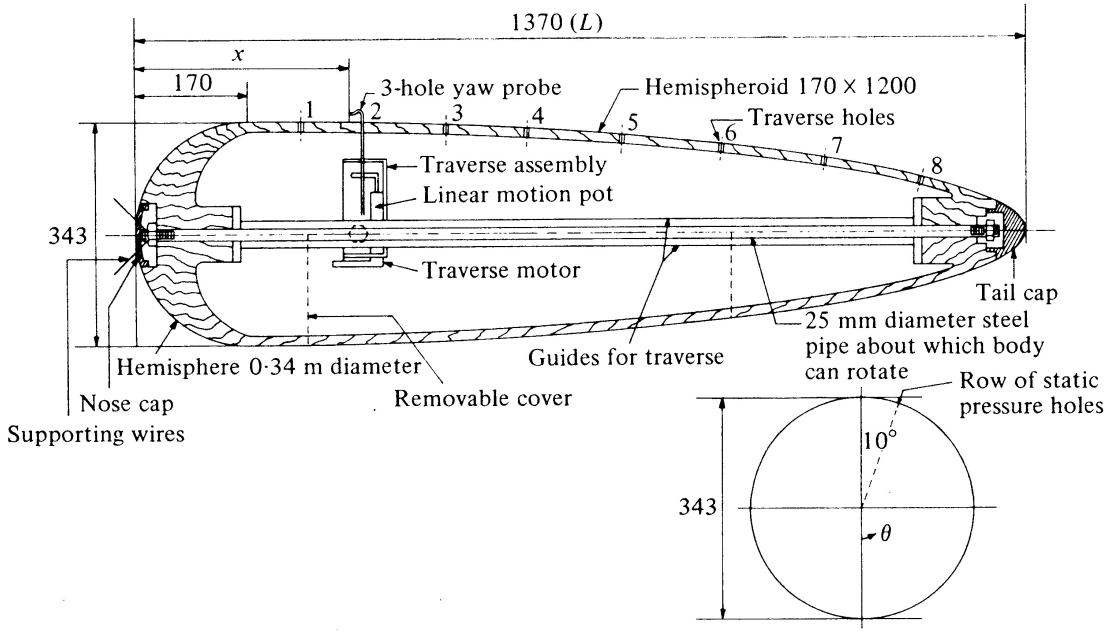


Figure 30 : Geometry and probe locations of the hemisperoid [70]

Table 5 : Traverse stations for the measurement of velocity profile and skin friction

Stations	1	2	3	4	5	6
x/L	0.176	0.241	0.333	0.426	0.537	0.648

The computed pressure distribution is shown in Figure 32, compared with the experimental data. The pressures are measured at circumferential angles of 0 and 120 degrees. The difference between the measured pressures is very small, and the computational

results show very good agreement with them. In Figure 33, calculated skin friction is compared with the experiments, in which the skin frictions are measured at the circumferential angles of 0 and 180 degrees. Since the flow is axisymmetric, the difference between these two is considered as an experimental error. According to the reference paper, the estimated error of the instrument in the wall shear stress measurement is 5 %. It is, however, reported that a considerable uncertainty exists due to the deviation of the angle between the velocity vector and the probe axis. Therefore, the error of the measured skin friction is assumed to be 10 % that corresponds to the difference of skin friction coefficients at station 6. The results of NASCART are well correlated to the experiment within the estimated error with the exception of the first measurement station. At station 1, the computed skin friction is overestimated, which also has been observed by Sondak and Pletcher [76]. They computed the turbulent flow over the same hemispheroid using three turbulence models (Baldwin-Lomax algebraic model, Launder and Spalding $k-\varepsilon$ model, and Chien's low Reynolds number $k-\varepsilon$ model), and none of the models predicts the proper skin friction at station1. This discrepancy would be caused by the transition from laminar to turbulent flows in the experiment, which is not modeled in numerical analysis. This would explain the fact that the measured skin friction coefficient is lower than the computed values and that the computation over-predicts the tangential velocity at station 1 in the subsequent plot. The computed velocity distributions are presented in Figure 34, and the results compare reasonably well with the test data. The small deviations at the stations 5 and 6 are due to the limit of the developed wall function boundary condition. As explained earlier, enforcing linear tangential velocity at the ghost cell may result in non-physical velocity at the boundary

cell. Excluding the boundary cell, predicted velocity profiles are well correlated with the experiments.

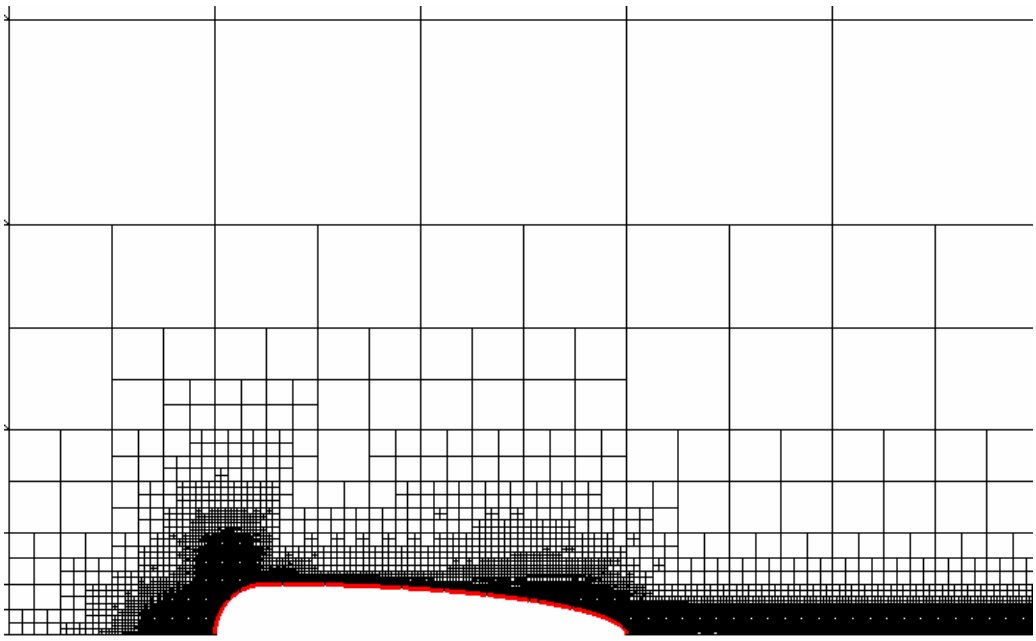


Figure 31 : Final grid configuration over hemispheroid

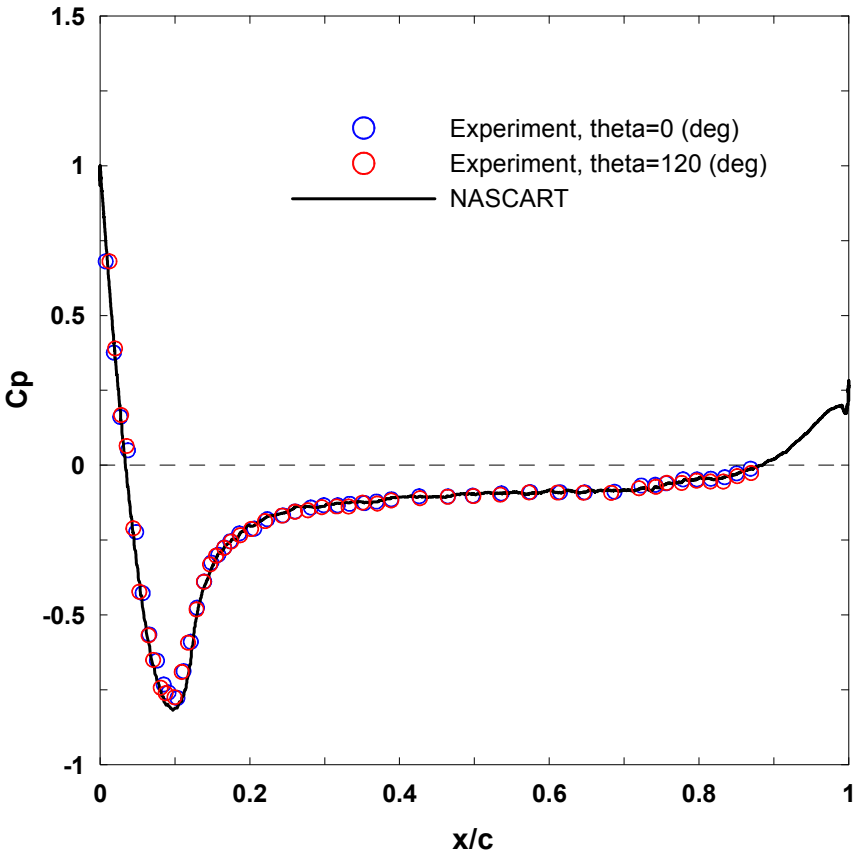


Figure 32 : Computational result of pressure coefficient over hemispheroid

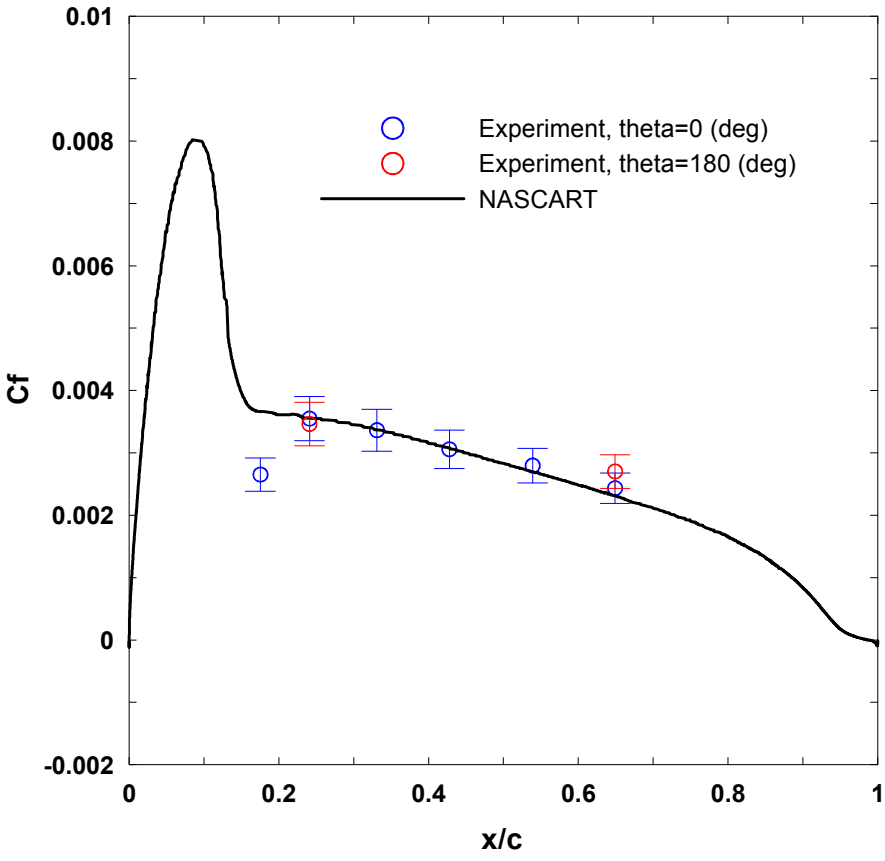


Figure 33 : Computational result of skin friction coefficient over hemispheroid

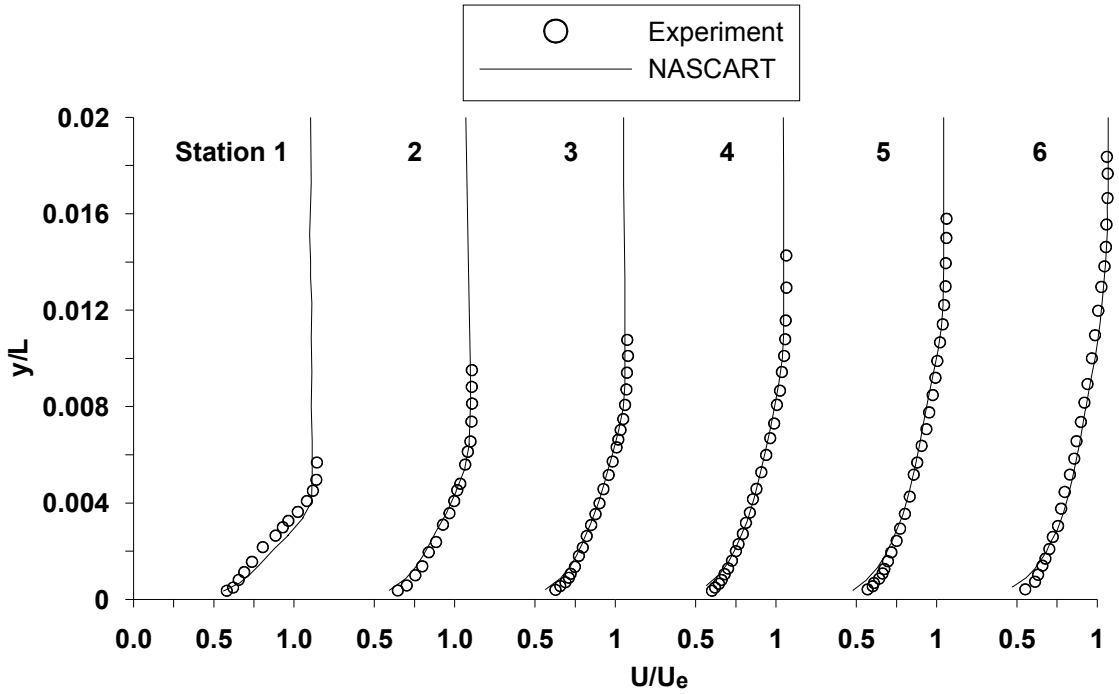


Figure 34 : Comparison of mean velocity profiles in flow field around hemispheroid

5.1.4 GRID EFFICIENCY

Although the law of the wall is valid even after the outer law commences (i.e. $y^+ > 300$), the traditional wall function approach is limited to much smaller y^+ as explained in Chapter III. In the present study, the grid dependency of the developed wall function boundary condition is tested by decreasing the maximum refinement level for the same root cell dimensions. This will result in the increase of boundary cell size, hence the increased y^+ near the wall.

Figure 35 presents the computed skin friction over flat plate. The computational parameters are identical to the case above, except that refinement level for the coarse grid is

8. It is easily noted that the difference between the results of the computations and experiments near the leading edge is decreased as the y^+ decreases. It is expected that the computed skin friction will further approach to the experimental data with higher grid resolution. In spite of the small error near the leading edge caused by insufficient grid resolution, the solution at $y^+ = 384.6$ shows good agreement with the results of fine grid computation and experiment at the trailing edge. This is also true for the solution of turbulent NACA 0012 airfoil case shown in Figure 36. The specifications of $y^+ = 81.1$ case are described above. The other solutions are obtained using the same numerical settings except the refinement levels. The blue line and the red line represent the solutions using the maximum grid refinement levels of 9 and 8, respectively. Computed Cp's of $y^+ = 81.1$ and $y^+ = 154.2$ are nearly identical, and that of $y^+ = 287.2$ is also very close to the other solutions and the test data with the exception of suction peak region. There exists a large pressure gradient near the suction peak region where the grid is not sufficiently resolved. This would dissipate the local peak and smear out the local gradient as exposed in the plot. This trend is also shown in the solutions of the flow around the axisymmetric hemispheroid in Figure 37 and Figure 38. The maximum refinement levels of the three cases are 11, 9 and 8, respectively. The computational results with medium and coarse grids ($y^+ = 154.6$ and $y^+ = 292.0$) underestimate the pressure and skin friction at $x/L \approx 0.1$ due to the numerical dissipation. It is also observed that the distributions of the skin friction become smoother as the maximum y^+ increases. However, the solutions of NASCART show reasonable correlations with the measured values within specified error limits excluding high gradient

regions of pressure and skin friction. In addition, the use of coarse grid system with currently developed wall function approach is useful in the initial estimation through the accurate prediction of pressure distribution, which plays major role in the aerodynamic forces of an arbitrary blunt body. This justifies the use of coarse grid system in the analysis of the rotorcraft flow field, in which the blunt fuselage interacts with the flows induced by the rotor.

For the quantitative analysis of the grid efficiency, the effects of the cell sizes on the accuracy and computational cost from the solutions of NACA 0012 airfoil flows are presented in Table 6. Errors are calculated based on the airfoil data [2] and the savings on the solutions of the RANS with the fine grid ($y^+ = 81.1$). There is little difference between computed lift coefficients for fine and medium grids whose errors are less than 0.2 %. Meanwhile the savings of computer memory and computation time are observed by over 48 % and 62 %, respectively. As mentioned before, conventional wall function approach requires small cells near wall to yield $y^+ < 80$ for proper turbulence modeling. This demonstrates the fact that the developed wall function approach has better grid efficiency and computational time than the conventional methods. Considering the coarse grid case, the accuracy is deteriorated just by 2.813 % while the computational time is improved by over 93 % with respect to the fine grid solution and over 81 % to the medium grid solution. It is, therefore, apparent that the implementation of coarse grid with the developed wall function method is practical and efficient approach in the overall prediction of the aerodynamic forces, yielding reasonable solution within the accuracy of the present turbulence model. It may be insisted that the Euler calculation is more efficient and accurate in an attached flow at high Reynolds number. The Euler calculations using the same grid resolutions with the fine and

coarse grids, however, over-predict the lift over 15 % as shown in the table, while the difference of CPU time in the Euler and RANS calculations with coarse grid is just 2.0 %. There are many possible reasons among which the displacement thickness plays important role in the error. In reality, the growth of the boundary layer affect the flow field as the thickness of the airfoil is increased. This would reduce the lift of the airfoil. Therefore, the viscous calculation is required to estimate the accurate aerodynamic forces even in an attached flow over a thin airfoil.

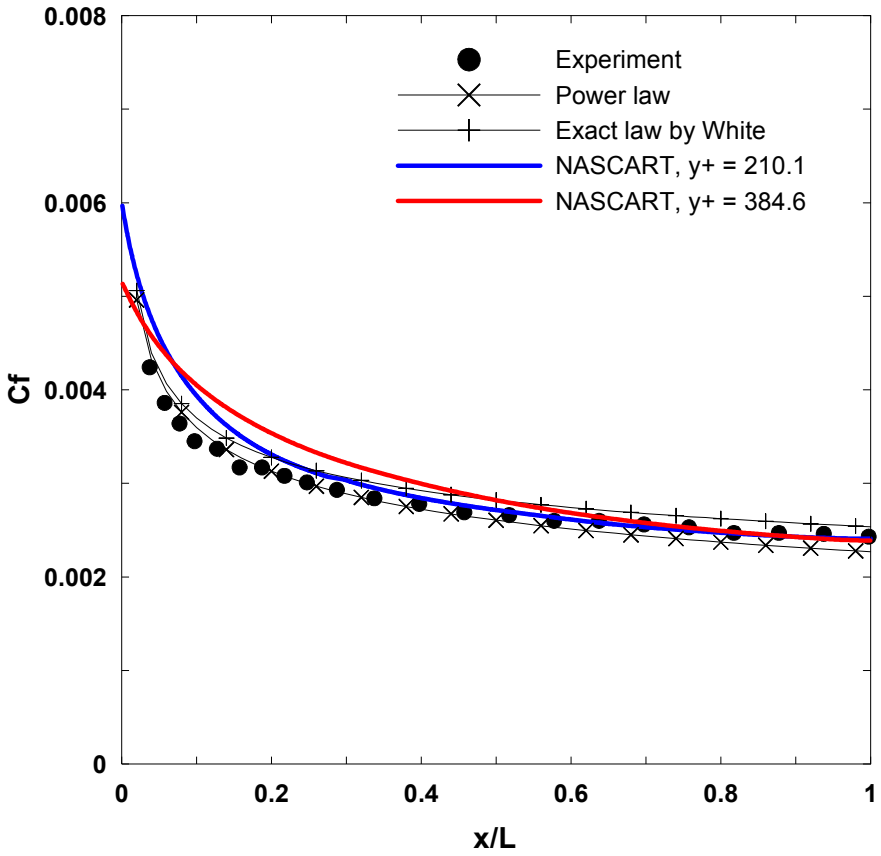


Figure 35 : Skin friction coefficients with various cell sizes over turbulent flat plate

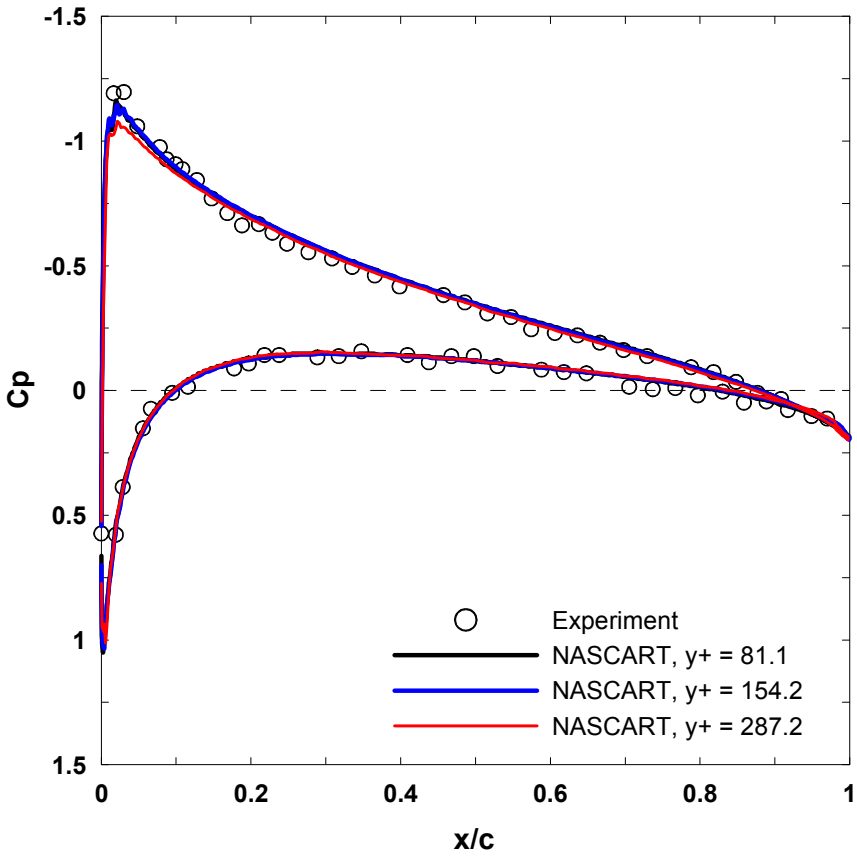


Figure 36 : Pressure coefficients with various cell sizes over turbulent NACA 0012 airfoil

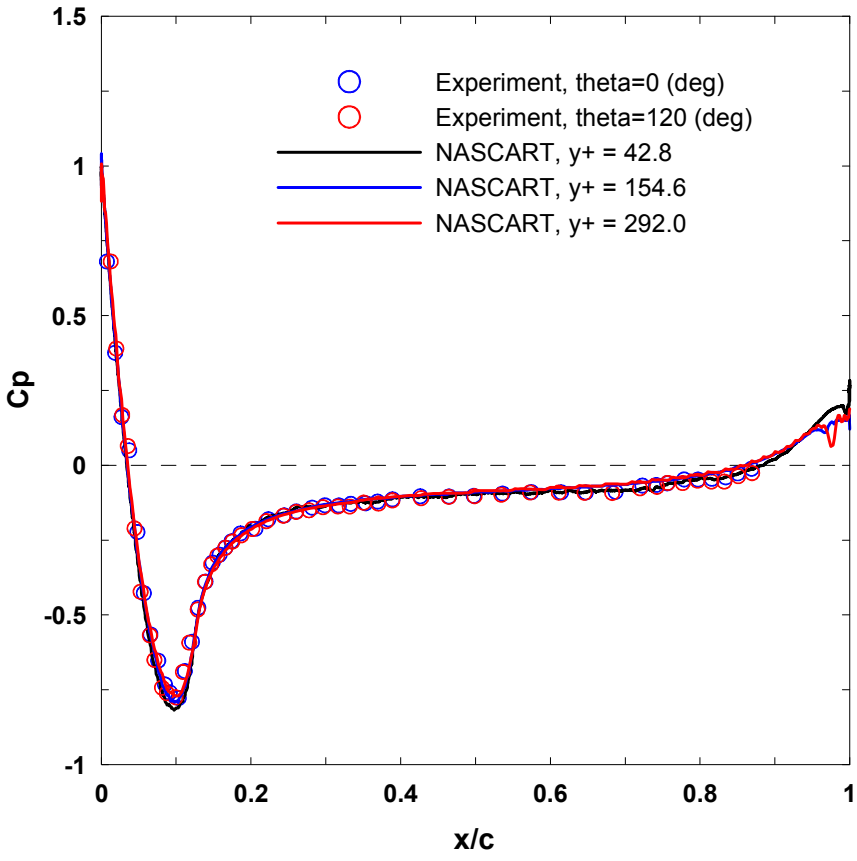


Figure 37 : Pressure coefficients with various cell sizes over hemispheroid

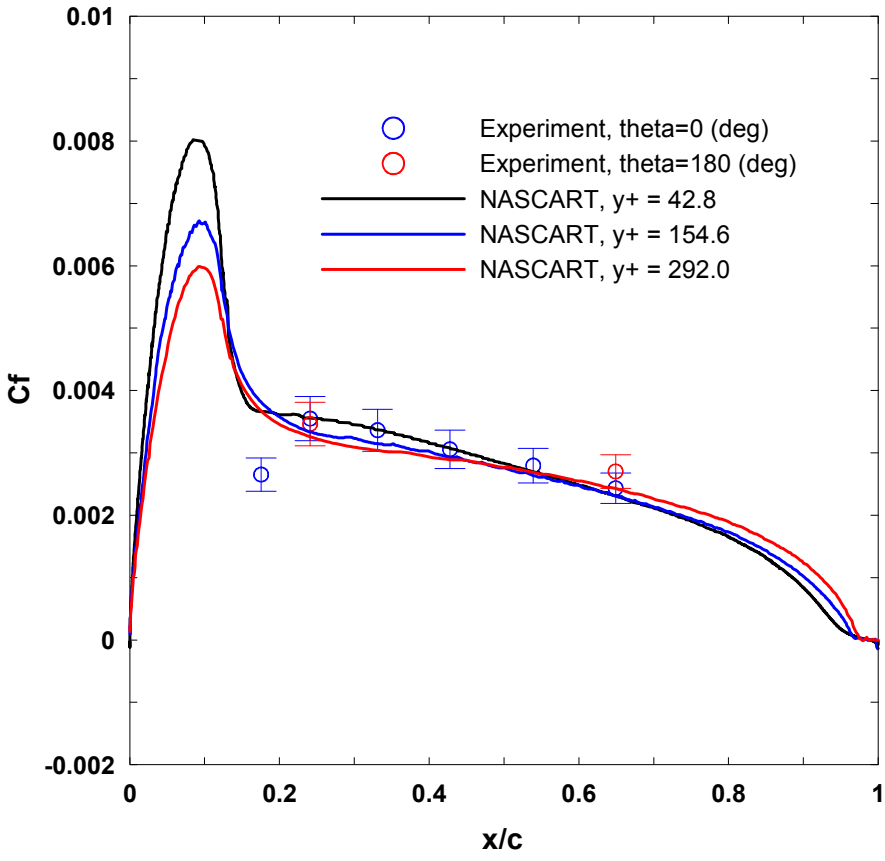


Figure 38 : Skin friction coefficients with various cell sizes over hemispheroid

Table 6 : Effect of cell size on accuracy and computational cost for NACA 0012 airfoil calculations

Solver, Grid density		RANS, Fine $y^+ = 81.1$	RANS, Medium $y^+ = 154.2$	RANS, Coarse $y^+ = 287.2$	Euler, Fine	Euler, Coarse
Accuracy	c_l	0.3584	0.3583	0.3489	0.4156	0.4185
	[+]Error (%)	0.1671	0.1950	2.813	15.77	16.60
Computer memory	No. of cells	72,520	37,672	20,176	15,058	4,474
	[++]Saving (%)	N/A	48.1	72.2	79.2	93.8
Computation time	* CPU time (hr)	44.7	16.9	3.1	37.3	2.2
	[++]Saving (%)	N/A	62.2	93.1	16.6	95.1

[+] Based on the airfoil data of $c_l = 0.3590$

[++] Based on the RANS fine grid calculation

* CPU time is obtained using Pentium IV 2.4 GHz PC

5.2 3-D ROTORCRAFT MODELS

For the numerical investigation of rotor-fuselage interaction, two configurations are selected. The first is Georgia Tech (GT) rotorcraft model [9], which is geometrically simple and has been extensively studied. The second application is ROBIN airframe model that has been tested by Freeman and Mineck [28]. Both of the configurations are computed without rotor model first for validation of developed code. Then, the rotor model using actuator disk is tested.

5.2.1 GT ROTOR MODEL

Figure 39 shows the configuration of GT rotor model. The fuselage consists of hemisphere and cylinder body. Two rotor blades have a 2.7% cutout and rectangular planform of NACA0015 section. Each rotor blades are untwisted with a constant pitch angle of 10 degrees and zero cyclic pitch angle. The clearance between rotor and airframe, h/R, is 0.3. Since the fuselage length is not specified, it is assumed to three rotor radii. In the experiments, the fuselage is mounted on a strut, which is not modeled in the computation due to the lack of detail geometry. The freestream is parallel to the fuselage, and the rotor shaft is tilted 6 degrees. The measured flap angle is shown in equation (5.4) without pre-cone.

$$\beta = -2.02^\circ \sin\psi - 1.94^\circ \cos\psi \tag{5.4}$$

where β is a flap angle, and ψ an azimuth angle. The rotational speed of the rotor is 2100 RPM with an advance ratio of 0.1. The measured thrust coefficient, C_T, is 0.009045. The corresponding Reynolds number based on the fuselage length is 9.196×10^5, and freestream

Mach number 0.029, assuming standard atmosphere at sea level. In the computation, the freestream Mach number is increased to 0.3 to prevent instability caused by incompressibility, while other nondimensional parameters (i.e. Reynolds number, thrust coefficient, and advance ratio) are maintained.

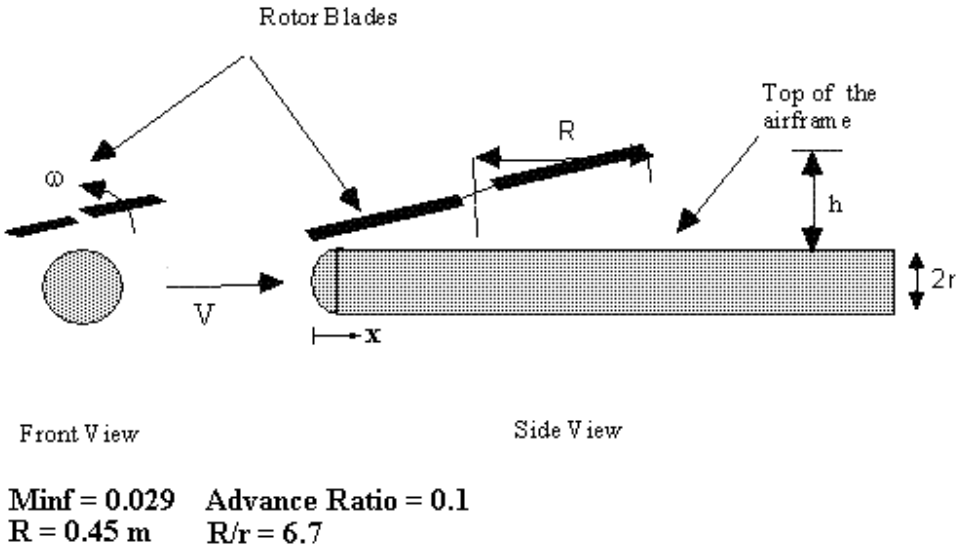


Figure 39 : Georgia Tech (GT) rotor configuration

In the present study, three numerical cases are tested on the GT rotor model. First, the Euler and the RANS calculations are performed without accounting for a blade flapping motion, which is intended to investigate the viscous effect on the rotor-fuselage interaction. The other case considers the effect of the blade flapping in the RANS calculation. Two methods of the blade flap motion in the actuator disk method are known to date. One adds the

flap velocity into the induced velocity normal to the disk plane that is located on the shaft plane [102, 103]. In the other method, the rotor disk is placed on the TPP without velocity modification [64]. In current study, the latter is chosen due to the difficulty in velocity scaling. A simple time derivative of the flap angle would not be suitable for specifying the disk boundary condition, since the freestream condition of the computation is different from the experiment. In addition, the flapping model using the TPP does not require an additional computation in the blade element method to find the flap velocity. In all three cases, the numerical boundaries are located 5 times of fuselage length from the center. The refinement level is 8 with a root cell dimension of $26 \times 22 \times 20$, which results in the largest y^+ of 169.9 and 218.6 for RANS calculations with and without flapping, respectively. The flow cells are refined at high gradient regions of divergence and vorticity. In the Euler calculation, the solution adaption is performed based only on the divergence. The total numbers of cells in the Euler and RANS calculations without flapping are 1,622,670 and 1,616,738, respectively, at the final iteration. That of the RANS calculation with flapping is 1,599,970. The freestream turbulent intensity is assumed to be 1% and the normalized freestream eddy viscosity is 0.1. Assuming smooth wall on the rotor fuselage, the parameter related to the roughness parameter is set to 5.0 utilized in the law of the wall.

The rotor disks are located on the shaft plane in the Euler and RANS calculations. The sectional lift and drag applied in the blade element method are shown in Figure 40. The maximal and minimal lift coefficients are assumed to be 1.6 and -1.6 at $\alpha = \pm 16°$, which correspond to the airfoil data in the reference [2]. The drag coefficient is obtained from the following equation.

$$c_d = 0.0087 - 0.0216\alpha + 0.4\alpha^2 \tag{5.5}$$

The constants used in the above equation are cited from Gessow and Tapscott [31]. The compressibility correction of the blade element method is not applied due to the low Mach number of the experimental model. The rotor trim analysis is performed at very 100 iterations to match the experimental rotor thrust. Since the GT rotor model has zero lateral and longitudinal cyclic pitch, only the collective pitch angle is adjusted by the trim loop.

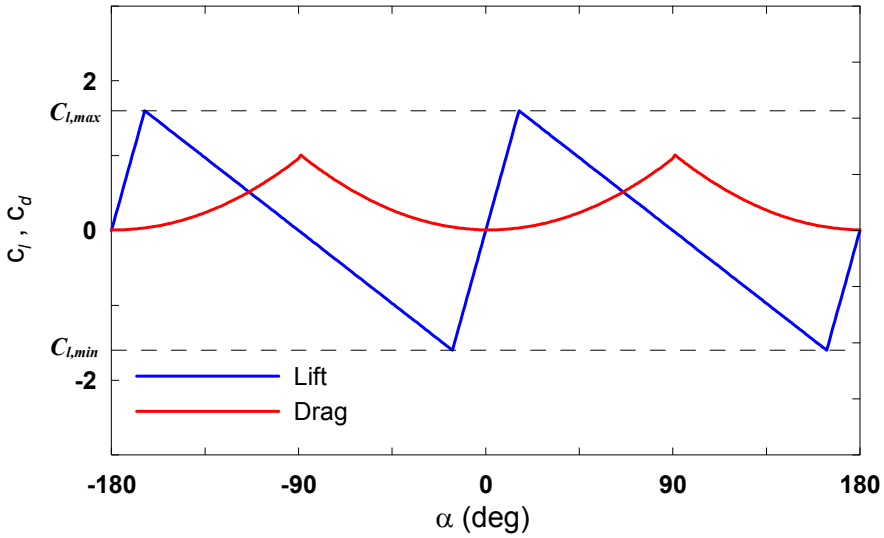


Figure 40 : Sectional lift and drag distributions in blade element method

The final grid configuration of a RANS calculation is shown with entropy contours in Figure 41. The transparent red disk represents the actuator disk. Considering that the quantity of entropy is a measure of vorticity, the flow cells near the vortex core are automatically

refined due to large vorticity. The cells near the rotor disk and the solid wall are also refined as a result of large divergence and vorticity, respectively. A close observation of the picture reveals that the vortex rollup generated at the tip of the rotor disk propagates and descends downstream. The tip vortex generated at the fore part of the disk propagates as a vortex sheet and dissipates rapidly, while the vortex generated at the lateral tips merges to a strong line vortex and travels further than the vortex sheet. This is clearly illustrated in Figure 42, which

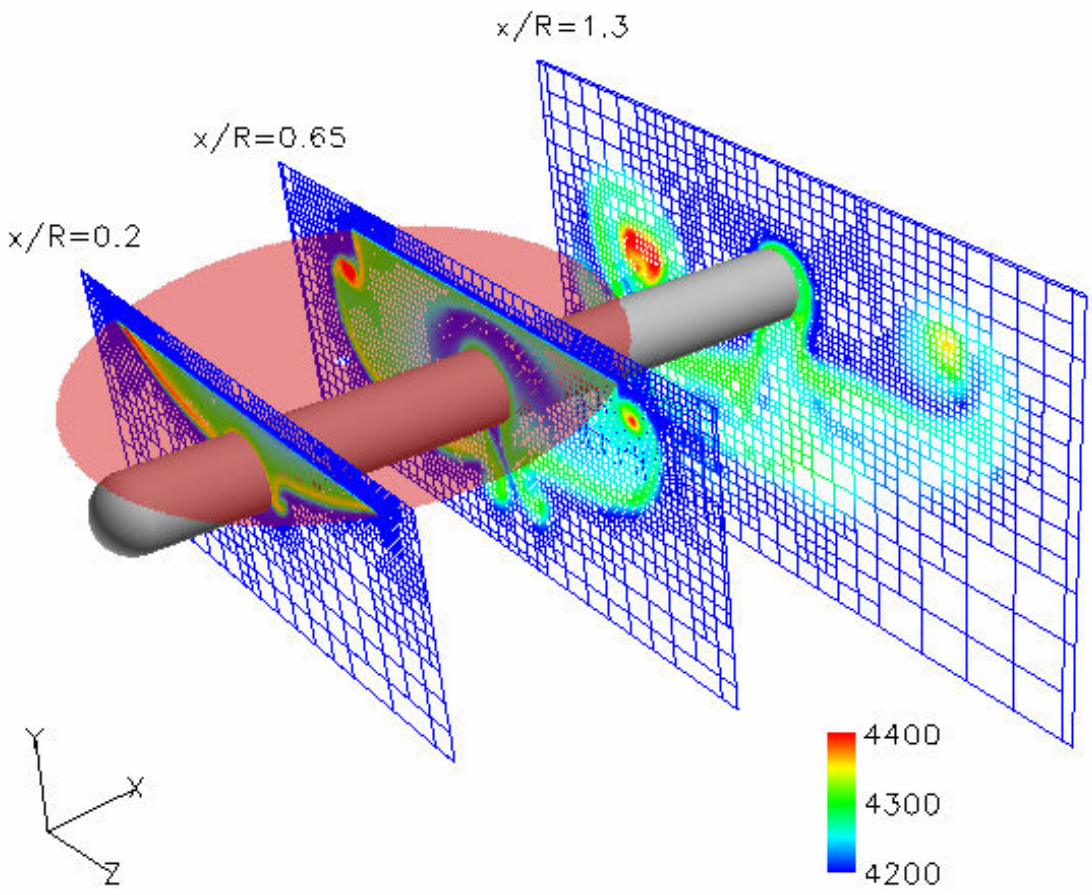


Figure 41 : Final grid configuration and entropy contours of GT rotor model from RANS calculation without flapping

shows iso-surfaces of the computed entropy. Note that the advancing side has a much stronger vortex rollup than the retreating side, since the rotor has a constant pitch angle.

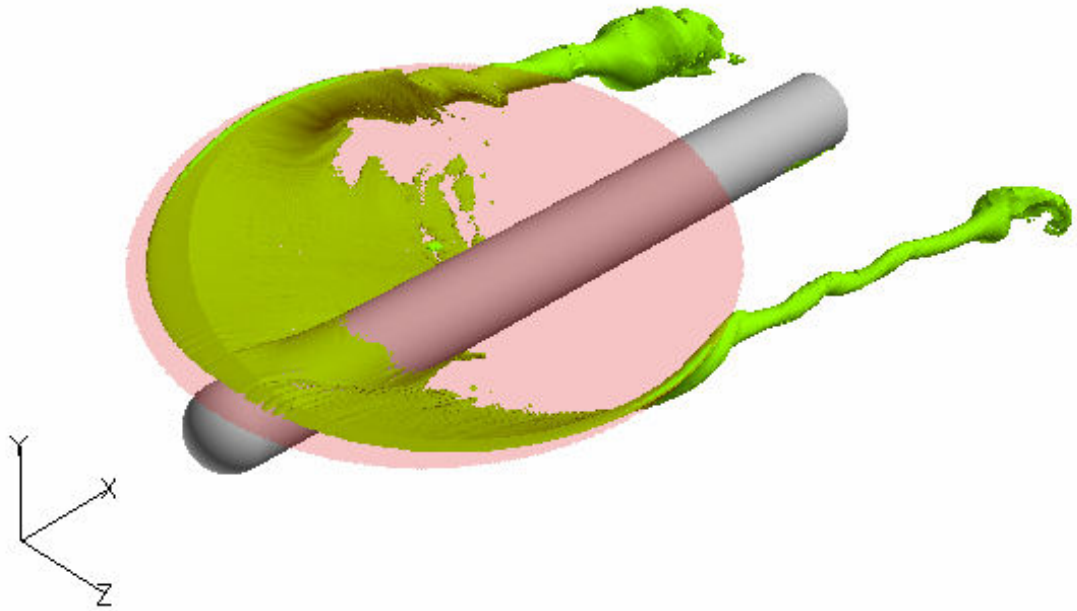


Figure 42 : Entropy iso-surfaces of GT rotor model from RANS calculation without flapping

The pressure distributions on the upper centerline of the GT rotor are shown in Figure 43. The Euler and the RANS results without flapping are very close to each other, and clearly show the peaks and drop of the pressure caused by the rotor motion. The solution with flapping motion shows much better correlation with the experiment and better accuracy in the prediction of the peak pressure than the other solutions. Considering the fact that the current flap model does not require an additional computation, the accuracy can be easily improved

without extra cost. However, there still exists an inconsistency with the measured pressure near the nose and the local peak at $x/R = 0.3$. The predicted pressure near the fuselage tail is also lower than the experiment. Because many researchers [50, 64, 66] also found these flow patterns in their actuator disk modeling on the GT rotor, it seems that the discrepancies in the computed pressure are not due to a problem of the current solver, but a limit of the actuator disk model. As shown in the graph, the problems of the under-prediction of the peak pressures and their locations can be solved by adding the rotor flapping, but the local peak problem at $x/R = 0.3$ still remains. In the experiment, the local peak seems to be primarily caused by the unsteady wake effect of the rotor blade motion. An individual rotor blade generates a number of vortex filaments, which interact with the wakes from the other blades. Therefore, the resultant wake appears to form helical line vortex, not to be vortex sheet as shown in Figure 42. A series of line vortices would be imposed on the surface, which drops the pressure. However, the actuator disk model is based on the time-averaged airfoil loading, which would not generate the helical shape line vortices but do the vortex sheet that dissipates rapidly. As explained above, the line vortex is stronger and propagates further than the vortex sheet. For more accurate calculation, a full unsteady computation might be required. This problem is analyzed and described well by O'Brien and Smith [64] who have extensively studied the rotor blade modeling. There is very small difference between the Euler and RANS results without flapping motion, excluding the fact that the RANS calculation results in slightly lower pressure due to the viscous dissipation and wake where the pressure peaks exist. This explains why other researchers [50, 66] employed an Euler

solver to analyze the GT rotor configuration. The trimmed collective pitch angles are presented in Table 7, which are within 3% error from the test result.

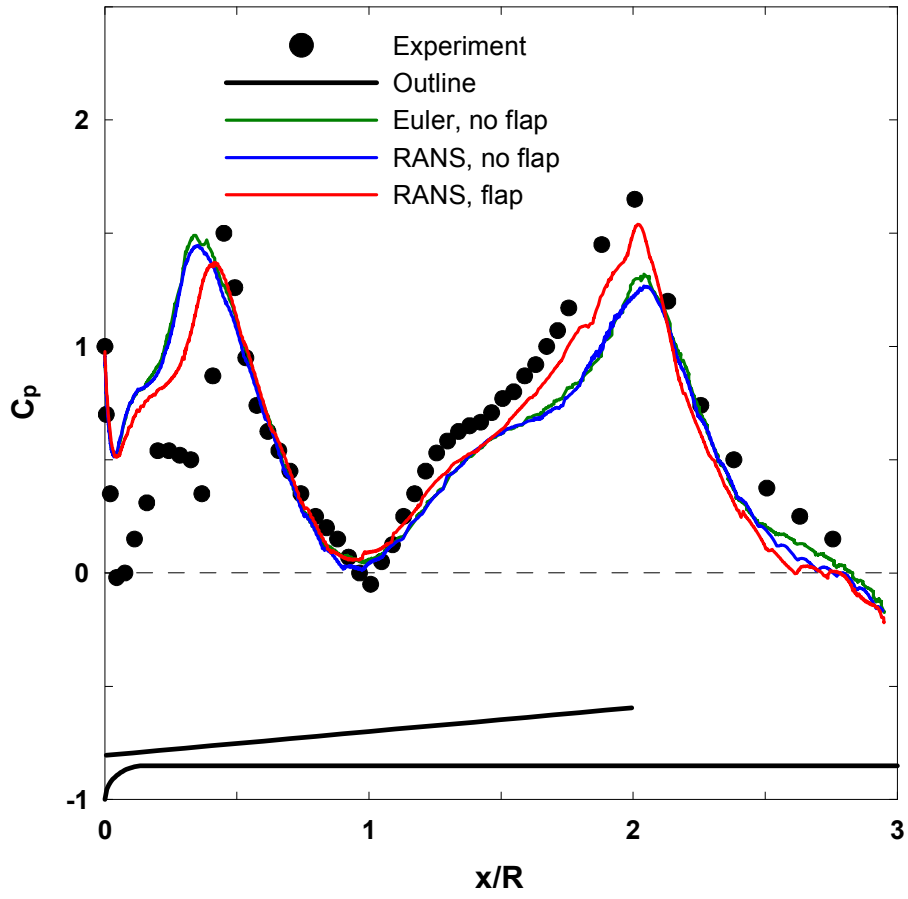


Figure 43 : Upper centerline pressure distribution for GT rotor model with rotor

Table 7 : Trimmed collective pitch angles in GT rotor model

	Experiment	Euler, no flap	RANS, no flap	RANS, flap
A_0 (degree)	10.0	9.879	9.856	9.720

5.2.2 ROBIN ROTORCRAFT MODEL

The Euler and RANS calculations using actuator disk are performed on the ROBIN (ROtor Body INteraction) model, which has been tested in NASA Langley in 2000 [58]. The fuselage shape is close to real helicopter, while its body is streamlined without any attachment for simplicity. The fuselage has the length of $2L$ where L represents the characteristic length. The fuselage centerline is yawed 1.2° nose left. The strut which supports the fuselage is mounted, is not modeled in the present study, since the detail geometry is now known. The model axes and sign conventions are shown in Figure 44. C_m and C_n represent the moment coefficients. C_A, C_N and C_Y denote axial, normal and lateral body force coefficients, respectively. The rotor consists of four blades, whose root cutout are at 24% of the radius, R, which is set to $R/L = 0.86$. The rotor blades have a rectangular planform with a chord of $0.066L$ and a linear twist of $-8°$. The center of the rotor hub is slightly offset to the advancing side, located at $x/L = 0.696$, $y/L = 0.051$, and $z/L = 0.322$.

The computations are performed to match the test conditions of $\mu = 0.051$. Assuming standard air at sea level, the freestream Mach number and the Reynolds number based on the fuselage length are 0.0266 and 1.312×10^6, respectively. The rotor shaft angle of attack is zero, and the measured thrust coefficient is 0.00636. The pitch angle of the rotor is given as

$$\theta = 5.9° + 1.3° \cos\psi - 1.3° \sin\psi + 8° \left(0.75 - \frac{r}{R} \right) \tag{5.5}$$

It is reported from the experiment that there is no significant pitch-flap coupling. Hence, the tip-path-plane is located normal to the rotor shaft and the coning angle is assumed to be zero.

In both Euler and RANS calculations, the freestream Mach number is increased to 0.3, while the other nondimensional parameters are matching with the experiment.

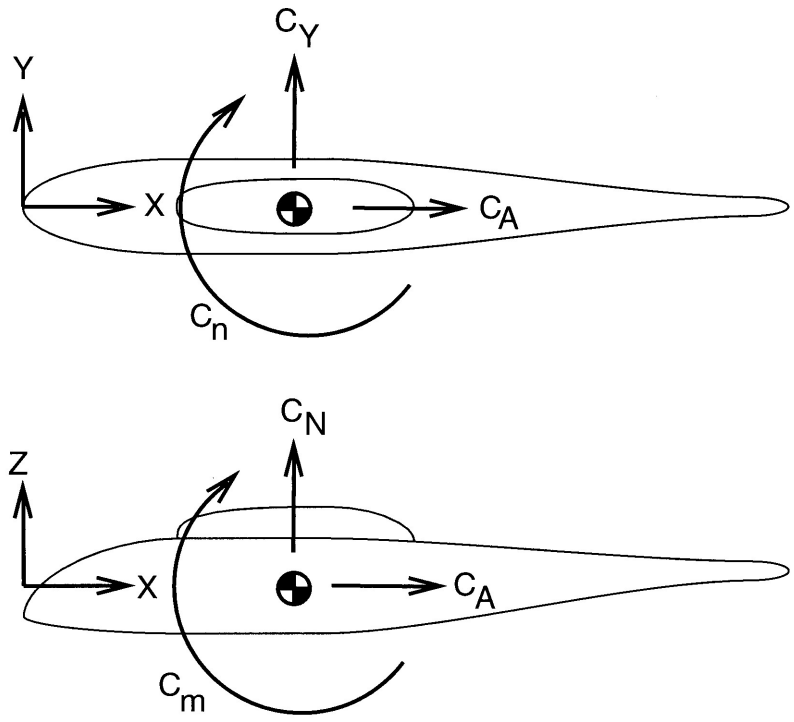


Figure 44 : Axes and sign conventions in ROBIN model from R.E. Mineck and S.A. Gorton [58]

The numerical boundaries are located 5 times of fuselage length from the center. The refinement level is 8 with a root cell dimension of $22 \times 16 \times 20$ to yield the maximum y^+ of 324.9 for RANS calculation. The Euler calculation also employs the same initial grid. The total numbers of cells in the final solutions are 740,377 and 1,614,516 for the Euler and the RANS calculations, respectively. The freestream turbulent intensity is assumed to be 1% and the normalized freestream eddy viscosity is 0.1. Assuming smooth wall on the rotor fuselage,

the parameter related to the roughness parameter is set to 5.0 utilized in the law of the wall. The solution adaption is performed based on the divergence and vorticity as done in GT rotor calculation. The actuator disk is located on the rotor shaft plane. Since the airfoil section is not given, the lift and the drag in the blade element method follow the GT rotor case. The compressibility correction is not applied due to the low Mach number of the experimental model. The rotor trim analysis is performed at very 100 iterations to match the experimental rotor thrust and to eliminate the lateral and longitudinal moments about the hub.

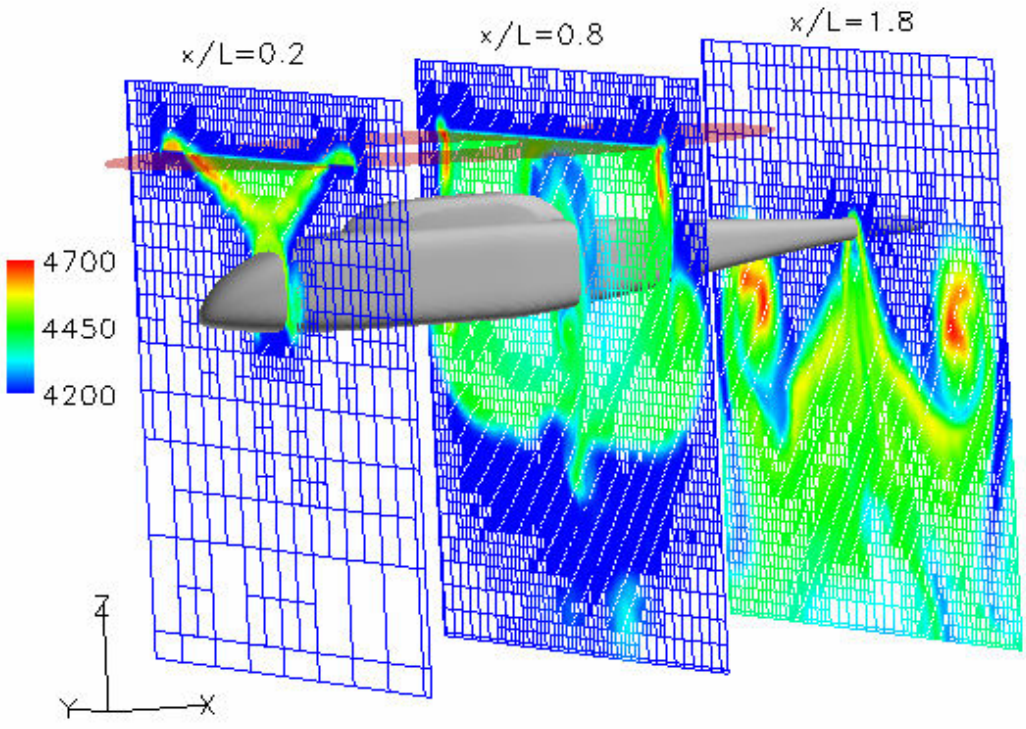


Figure 45 : Final grid and entropy contours of ROBIN configuration

The adapted grid configuration and entropy contours are shown in Figure 45. As shown in the GT rotor case, strong line vortex is formed at lateral sides of the rotor disk and propagates downstream. The cells are refined along with the propagated wakes. Relatively low entropy region is shown at $x/L = 0.8$ in a doughnut shape, which is caused by the root cutout of the rotor. Since there is no blade loading inside the cutout region, the increase of the entropy is smaller than its surrounding.

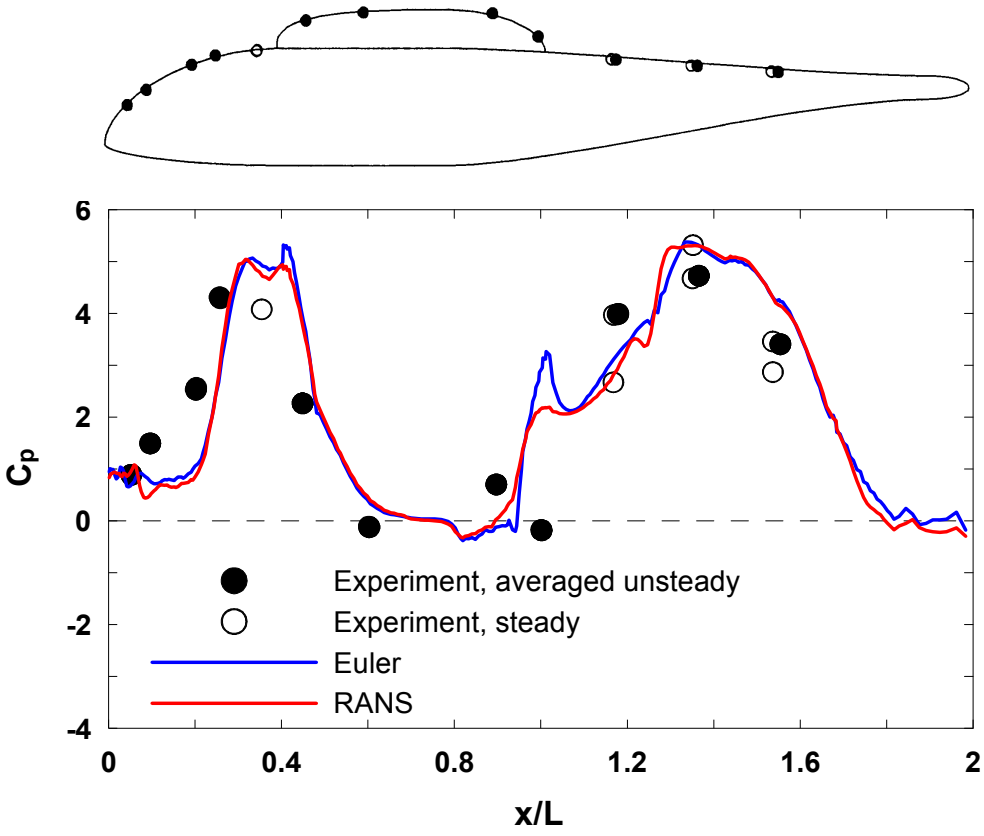


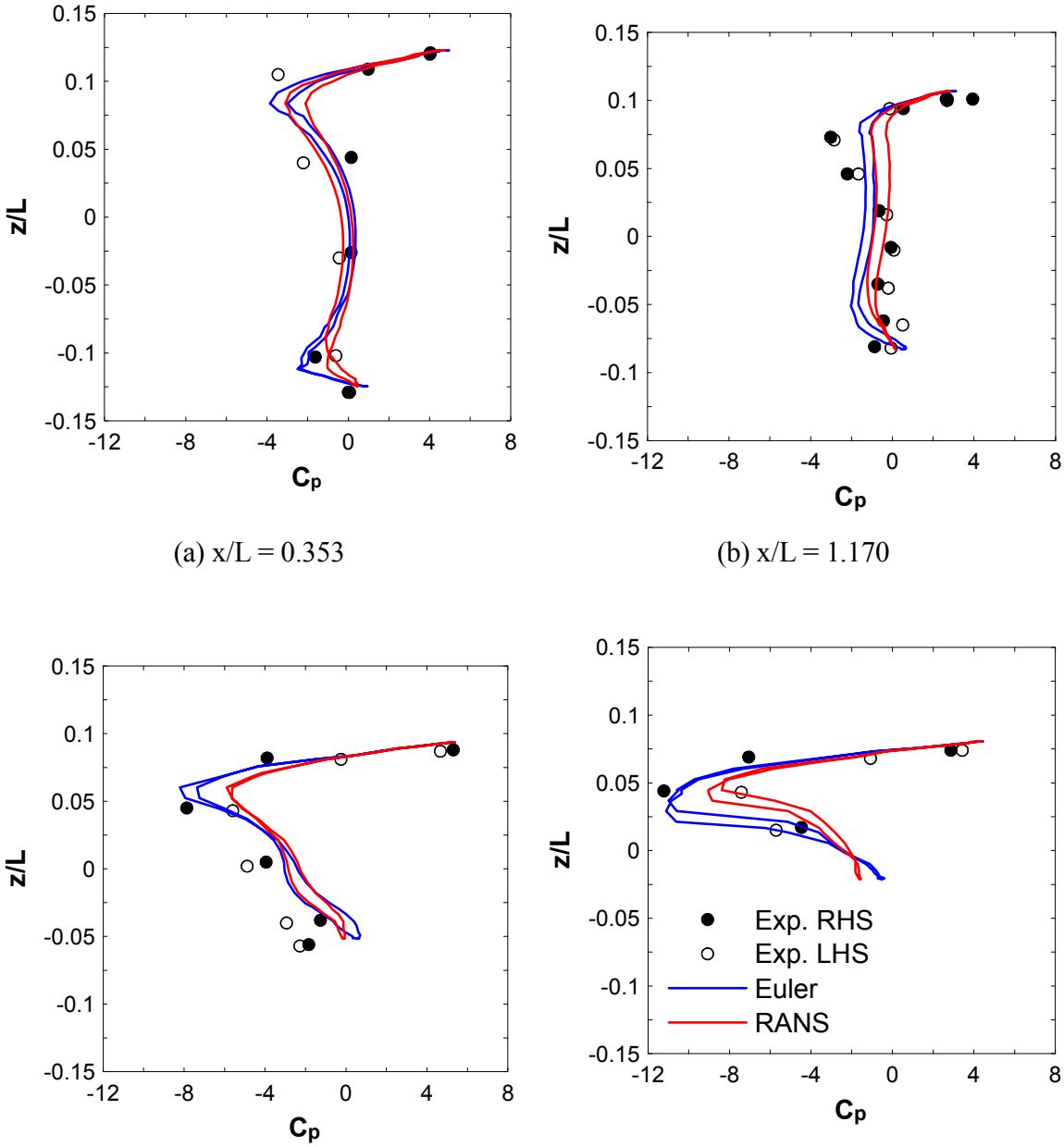
Figure 46 : Pressure distribution on the upper centerline of ROBIN configuration

The comparison of the computed pressure distribution on the upper centerline is presented in Figure 46. The open symbols represent the steady state pressure, and the filled symbols represent the averaged value from unsteady pressure. Since the measured locations have slight offsets from the upper centerline, two values are plotted at the same x-locations. Both of the Euler and the RANS results are well correlated with the measured values, and there is not a large difference between them except the regions around the pylon and the nose. The difference between two numerical solutions occurred near the pylon is appears to be due to flow separation. The Euler solver can not simulate the flow separation. This produces the local pressure peak, which disappears in the RANS solution.

The sectional pressure distributions at various x-locations are shown in Figure 47. The solutions from the Euler and the RANS calculations are similar and show good agreement with the experiment near the nose at $x/L = 0.353$. The difference between two solutions is increased as the sectional location approaches downstream. This seems reasonable since the boundary layer grows downstream and the difference between inviscid and viscous solution would be enlarged. However, the numerical solutions do not clearly distinguish the pressure on the lift and right surfaces and follow the measured value of one side at the tail region as shown in the picture (d). One of the possible reasons is the difference of the model configurations between the computations and the experiment. In the wind tunnel test, there exist a rotor shaft above the pylon and a strut underneath the fuselage at $x/L \approx 1.0$, which is not reflected in the numerical analysis. These attachments would produce a complicated flow pattern that is different from the computational result. A close observation of Figure 47 (b) right aft of the strut reveals that the measured pressure is decreased at the bottom while the

computed pressure is increased. In the experiment, the flow would separate around the strut and the resultant vortex would propagate downstream. This may affect the pressure field downstream and yield the discrepancy in Figure 47 (c). The rotor shaft is also likely to disturb the flow around the rotor disk, which alters the disk boundary condition and resultant disk loading. This may explain that the over-predicted pressure at the tail region in Figure 46 and a large difference between numerically trimmed and measured pitch angles in Table 8.

The effect of the strut can be seen in Figure 48, which presents computed Cp contours around the ROBIN fuselage without rotor. The Euler solutions of with and without strut are compared with the experimental data in reference [28]. The inclusion of the strut has noticeable effect on the Cp distribution near and aft of the mounting point, and the computation result including the strut shows better agreement with the measurements. The Euler solution of section pressure distribution at x/L=1.0008 is shown in Figure 49. The compared experimental data are obtained at Run 12 Point 90 and Run13 Point 94 in the reference. Distinct pressure drop near the bottom is observed with strut, which is not illustrated without strut in the same figure or in Figure 47 (b). A computation assuming the rotor shaft and the strut in the reference [64] also showed the effects of the attachments on the flow field. For more accurate computation, the detailed information of experimental model strut is required.



(a) x/L = 0.353

(b) x/L = 1.170

(c) x/L = 1.354

(d) x/L = 1.540

Figure 47 : Pressure distribution across ROBIN section

Table 8 : Trimmed pitch angles of ROBIN model

	Experiment	Euler	RANS
A_0 (degree)	5.9	10.12	10.05
A_1 (degree)	-1.3	-0.94	-0.98
B_1 (degree)	1.3	1.65	1.53

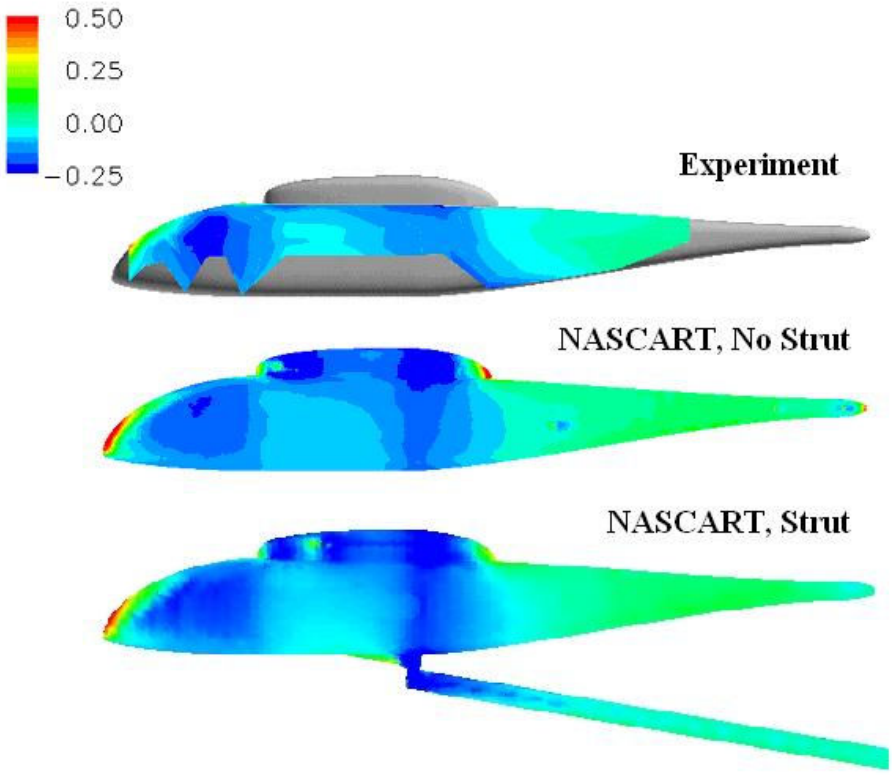


Figure 48 : Effect of strut on Cp contours without rotor

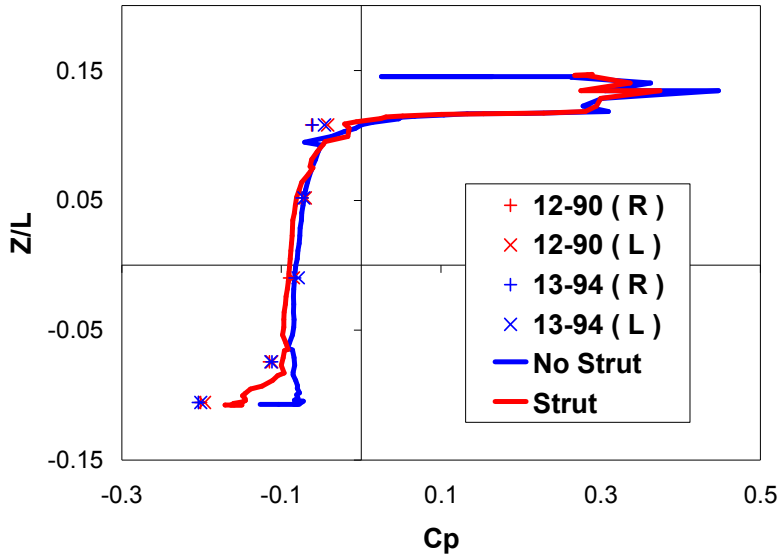


Figure 49 : Sectional Cp distribution without rotor

C H A P T E R V I

CONCLUSIONS AND EXTENSIONS

6.1 CONCLUSIONS

In present study, new wall boundary conditions are implemented into the existing unstructured Cartesian grid framework. Instead of an embedded wall boundary, immersed boundary approach is applied with ghost cell boundary condition. The centers of boundary cells in an immersed boundary approach remain at the centers of the uncut cells, while those in an embedded boundary approach are shifted to the centroids of the cut cells. By comparing the solutions of the immersed ghost boundary approach and an embedded non-ghost boundary approach, the effects of the aligned cell centroids and the conservative volume integration are investigated. The standard k-epsilon model by Launder and Spalding is employed in the calculation of RANS equations for the turbulence modeling, and new wall function approach is devised for the immersed unstructured Cartesian grid solver. The new boundary condition is verified in many 2-D flow calculations by comparing the solutions with measured data. The RANS solver with developed boundary condition is, then, applied in the analysis of the rotorcraft model with the actuator disk model to simulate the rotor-fuselage interaction. A number of conclusions are shown below.

- The boundary cell centroids aligned with the flow cell centers make the numerical stencil orthogonal and reduce the error in the volume integration. The application of

ghost cell approach also increases the accuracy by the use of conservative volume integration in the calculation of boundary cells just like in flow cell calculation instead of simple extrapolation of the flow properties. This yields an accurate prediction of skin friction and velocity profile in a boundary layer as well as the pressure distribution in the calculation of Navier-Stokes equations. Comparing the results with the embedded and immersed boundary approaches in the Navier-Stoke calculation, it is observed that the accuracy is improved and unphysical fluctuation is minimized.

- It has been observed that the conventional wall function approach of no-slip wall boundary condition induces a large magnitude of unphysical fluctuations of flow properties and instability of the computation when it is applied into the immersed Cartesian grid solver. The use of slip wall boundary condition results in unphysical transient solution, thereby, takes large computation time to get the converged solution.
- The new boundary condition is developed and successfully tested for an immersed Cartesian grid solver, which is never done before. The developed wall function approach yields stable and reasonable solution within the accuracy of the turbulence model. The new approach removes the complicated coordinate transformation required in the conventional wall function approach of either slip or no-slip wall function method, which reduces the truncation error and increases the accuracy.

- Unlike the conventional wall function approach, the developed method shows stable and reasonable solution with a relatively coarse grid system within the accuracy of the turbulence model. The use of a coarse grid with the developed wall function approach can reduce the computational memory and computation time.
- When the RANS solver with developed boundary condition is applied in the analysis of the rotorcraft model with the actuator disk model, the wakes created by the rotor disk are well captured using the adapted grid technique.
- The actuator disk model with the blade element method provides a good analysis of the steady state influence of the rotor in a couple in the computations of the GT rotor and the ROBIN model, although complicated unsteady effects may not be revealed. This yields a reasonable solution within the accuracy of the computational models used, and is computationally efficient (in terms of CPU time and memory on a single PC).
- The inclusion of blade flap into the actuator disk model improves the accuracy without an additional computation cost.
- Considering that the unstructured Cartesian grid solver has an advantage over the other grid topology in the grid generation over a complex geometry, the current research would provide a very useful aerodynamic tool in the preliminary design of a helicopter.

6.2 EXTENSIONS

The present work provides a solid basis of the viscous simulation using unstructured Cartesian grid solvers.

The most significant limit of the current solver is the memory and computation time in an analysis of three-dimensional viscous flow. A viscous simulation of complex geometry like a real helicopter requires a great amount of computer memory and CPU time. To extend the limit, the parallelization of the solver should be implemented. This will enable the unstructured Cartesian grid solver to be practically valid for aerodynamic design tool utilizing the capability of automated grid generation and grid efficiency over other unstructured grid topology. The accuracy of the calculation can also be improved further by applying more advanced turbulence models if necessary. For example, the unstructured Cartesian cells can be adapted depending on an eddy size without having any skewed cell, when the LES (Large Eddy Simulation) is applied to the solver. This would not only increase the accuracy of the simulation, but also save computer memory.

For the accurate calculation of unsteady phenomena in the rotor-fuselage interaction, either an overset grid approach or a moving grid technique is required. Unlike the overset grid approach, the moving grid method uses single computational frame. Therefore, more accuracy can be obtained by eliminating the interface between a stationary frame and a rotating frame. An immersed Cartesian grid approach has an advantage over other unstructured grid methods in the grid generation around the moving boundaries, since the cells are located independent of the solid wall boundary. The only difference in the treatment of a boundary cell from a flow cell is that the state vector of the neighbor cell is calculated

from the boundary condition. This simplifies the complicated and laborious moving boundary treatment. The moving grid technique along with the MPI (Message Passing Interface) would extend the limit of the current solver and yield very accurate solution of a rotor-fuselage interaction.

Another extension of the current solver would be to add an incompressible flow capability utilizing an artificial compressibility with preconditioning as describe in reference [91]. The greatest advantage of this method is the convenience of reconciling the compressible and incompressible solvers. The compressible governing equations can be easily converted to incompressible flow via preconditioning matrix, thereby the state vector and corresponding flux vectors have the similar format with the compressible flow solver. Without the preconditioning, one must use a different set of state and flux vectors for incompressible flow solver, which takes severe effort to make another solver. Implementing this approach has a great benefit in the simulation of the rotor-fuselage interaction using single computational frame and moving grid technique. Generally, the flow around rotating rotor blade is a compressible flow due to the large tip velocity, while the flow around the fuselage is assumed to be an incompressible flow due to low speed in forward flight. Applying a compressible flow solver may result in an instability caused by poor conditioning near the fuselage. Using an artificial compressibility with the preconditioning scheme, the compressible flow around the rotor and the incompressible flow around the fuselage can be properly simulated

A P P E N D I X A

INTERPOLATION METHODS

This appendix provides the details of interpolation methods used in the current solver. The variables on the numerical stencil are obtained using pseudo-Laplacian weighted averaging method, and those on the reference point of a boundary cell are calculated utilizing linear least square interpolation. These methods are described in the next two sections.

A.1 PSEUDO-LAPLACIAN WEIGHTED AVERAGING

As explained in chapter II, the state vectors at a given numerical stencil point are calculated by pseudo-Laplacian weighted averaging method from the neighbor cells. This increases accuracy and produces more robust convergence. N. T. Frink [29] applied this method on a three dimensional unstructured grid solver. However, its accuracy is known to be less than second-order. A fully second-order accurate averaging procedure was presented by D. G. Holmes [37] for two-dimensional triangular cells. In NASCART-GT, Holmes's approach is extended for three dimensional Cartesian cells. This section describes their formulations.

Estimates of the solution are determined at each point of interest by a weighted average of the surrounding cell centered solution quantities:

$$\boldsymbol{Q}_c = \left(\sum_{i=1}^{N} w_i \boldsymbol{Q}_i \right) / \left(\sum_{i=1}^{N} w_i \right) \tag{A.1}$$

The subscript c and i refer to the point of interest and its surrounding cell-centered values, respectively. N represents the number of neighbor cells. For inverse-distance weighting factor,

$$w_i = \frac{1}{r_i} \tag{A.2}$$

where

$$r_i = \left[(x_i - x_c)^2 + (y_i - y_c)^2 + (z_i - z_c)^2 \right]^{1/2}$$

This weighting has been used quite successfully for computing the flow over a number of complex configurations.

Pseudo-Laplacian weighted averaging was modified by D.G. Holmes to get a fully second-order accurate averaging procedure. The following formulation was extended from his two dimensional averaging to three dimensional one. The method is based on deriving weight factors in equation (A.2) which satisfy the Laplacians:

$$\begin{aligned} L(x_c) &= \sum_{i=1}^{N} w_i (x_i - x_c) = 0 \\ L(y_c) &= \sum_{i=1}^{N} w_i (y_i - y_c) = 0 \\ L(z_c) &= \sum_{i=1}^{N} w_i (z_i - z_c) = 0 \end{aligned} \tag{A.3}$$

This is a desirable property since the Laplacian of a linear function is exactly zero. The weights are determined by defining

$$w_i = 1 + \Delta w_i \tag{A.4}$$

where

$$C = \sum_{i=1}^{N} (\Delta w_i)^2$$

is a cost function. The cost function is minimized by solving an optimization problem subject to the constraints of equation (A.3). This optimization problem is solved by the method of Lagrange multipliers, where Δw_i is given by

$$\Delta w_i = \lambda_x (x_i - x_c) + \lambda_y (y_i - y_c) + \lambda_z (z_i - z_c) \tag{A.5}$$

The solution to the constrained optimization problem yields the Lagrange multipliers

$$\begin{aligned}
\lambda_x &= \left[-R_x (I_{yy} I_{zz} - I_{yz}^2) + R_y (I_{xy} I_{zz} - I_{xz} I_{yz}) - R_z (I_{xy} I_{yz} - I_{yy} I_{xz}) \right] / D \\
\lambda_y &= \left[R_x (I_{xy} I_{zz} - I_{xz} I_{yz}) - R_y (I_{xx} I_{zz} - I_{xz}^2) + R_z (I_{xx} I_{yz} - I_{xy} I_{xz}) \right] / D \\
\lambda_z &= \left[-R_x (I_{xy} I_{yz} - I_{yy} I_{xz}) + R_y (I_{xx} I_{yz} - I_{xy} I_{xz}) - R_z (I_{xx} I_{yy} - I_{xy}^2) \right] / D
\end{aligned} \tag{A.6}$$

where

$$D = I_{xx} (I_{yy} I_{zz} - I_{yz}^2) - I_{xy} (I_{xy} I_{zz} - I_{xz} I_{yz}) + I_{xz} (I_{xy} I_{yz} - I_{yy} I_{xz})$$

$$R_x = \sum_{i=1}^{N} (x_i - x_c)$$

$$R_y = \sum_{i=1}^{N} (y_i - y_c)$$

$$R_z = \sum_{i=1}^{N} (z_i - z_c)$$

$$I_{xx} = \sum_{i=1}^{N} (x_i - x_c)^2$$

$$I_{yy} = \sum_{i=1}^{N} (y_i - y_c)^2$$

$$I_{zz} = \sum_{i=1}^{N} (z_i - z_c)^2$$

$$I_{xy} = \sum_{i=1}^{N} (x_i - x_c)(y_i - y_c)$$

$$I_{xz} = \sum_{i=1}^{N} (x_i - x_c)(z_i - z_c)$$

$$I_{yz} = \sum_{i=1}^{N} (y_i - y_c)(z_i - z_c)$$

These weights are constructed entirely from geometric information.

A.2 LINEAR LEAST SQUARE INTERPOLATION

Least square interpolation is applied to obtain primitive variables at the reference point as explained above. This method is to solve an overdetermined linear equation set according to the minimum energy principle. The following shows the summary of its formulation. Check the references [84, 99] for detail.

The overdetermined system of linear equations are expressed as

$$\boldsymbol{A}\nabla W = \Delta W \tag{A.7}$$

where

$$\boldsymbol{A}=\begin{bmatrix}\Delta x_1 & \Delta y_1 & \Delta z_1\\ \vdots & \vdots & \vdots\\ \Delta x_N & \Delta y_N & \Delta z_N\end{bmatrix}\qquad \nabla W=\begin{bmatrix}\nabla_x W\\ \nabla_y W\\ \nabla_z W\end{bmatrix}\qquad \Delta W=\begin{bmatrix}\Delta W_1\\ \vdots\\ \Delta W_N\end{bmatrix}$$

N represents the number of neighbor points. The least square method is to find the pseudo inverse of matrix $\boldsymbol{A}$ such that

$$\nabla W=\boldsymbol{A}^{+}\Delta W \tag{A.8}$$

where the pseudo inverse matrix is

$$\boldsymbol{A}^{+}=\left(\boldsymbol{A}^{T}\boldsymbol{A}\right)^{-1}\boldsymbol{A}^{T} \tag{A.9}$$

The expressions for the gradients are shown below.

$$\begin{aligned}\nabla_x W&=\frac{1}{K}\sum_{i=1}^{N}\left[(c_2+c_3)(x_i-x_0)-c_4(y_i-y_0)-c_5(z_i-z_0)\right](W_i-W_0)\\ \nabla_y W&=\frac{1}{K}\sum_{i=1}^{N}\left[-c_4(x_i-x_0)+(c_1+c_3)(y_i-y_0)-c_6(z_i-z_0)\right](W_i-W_0)\\ \nabla_z W&=\frac{1}{K}\sum_{i=1}^{N}\left[-c_5(x_i-x_0)-c_6(y_i-y_0)+(c_1+c_2)(z_i-z_0)\right](W_i-W_0)\end{aligned} \tag{A.10}$$

$$K=c_1c_2+c_1c_3+c_2c_3-c_4^2-c_5^2-c_6^2$$

The subscript, 0, represents the base point of interpolation. The base point of interpolation could be one of the neighbor point or other point surrounded by neighbor point. For inviscid and laminar wall boundary condition, it is set to be the closest neighbor point from the boundary reference point. The geometric coefficients, c_1 through c_6, are defined as

$$c_1=\sum_{i=1}^{N}(x_i-x_0)^2$$

$$c_2 = \sum_{i=1}^{N} (y_i - y_0)^2$$

$$c_3 = \sum_{i=1}^{N} (z_i - z_0)^2$$

$$c_4 = \sum_{i=1}^{N} (x_i - x_0)(y_i - y_0) \tag{A.11}$$

$$c_5 = \sum_{i=1}^{N} (x_i - x_0)(z_i - z_0)$$

$$c_6 = \sum_{i=1}^{N} (y_i - y_0)(z_i - z_0)$$

Let the subscript, p, denote the point of interest or the point where to interpolate, then the interpolated value at the point, p, is expressed as

$$W_p = W_0 + (x_c - x_0)\nabla_x W + (y_c - y_0)\nabla_y W + (z_c - z_0)\nabla_z W \tag{A.12}$$

The wall boundary condition using the reference point employs this formulation to find the primitive variables at the reference point.

A.3 NEIGHBOR FINDING ROUTINE

Prior to interpolation, the Cartesian scheme requires that all neighbors of a given cell be identified. This information could, in theory, be stored in memory but is instead determined whenever needed. This saves memory but leads to greater CPU time. For 2-D cases, 13.7% of CPU time is spent in neighbor finding and for 3-D cases, 27.9% of CPU time is spent in neighbor finding routine.

BIBLIOGRAPHY

[1] "Experimental Data Base for Computer Program Assessment". AGARD Advisory Report No. 138.

[2] I.H. Abbott and A.E. Von Doenhoff, *Theory of Wing Sections*. Dover Ed., 1959.

[3] M.J. Aftosmis, M.J. Berger, and G. Adomavicius, "A Parallel Multilevel Method for Adaptively Refined Cartesian Grids with Embedded Boundaries". AIAA Paper 2000-0808, 2000.

[4] J.D. Anderson, *Modern Compressible Flow with Historical Perspective*. 2nd Ed., McGraw-Hill, 1990.

[5] B.S. Baldwin and T.J. Barth, "A One-Equation Turbulence Transport Model for High Reynolds Number Wall-Bounded Flows". AIAA Paper 91-0610, 1991.

[6] M.J. Berger and R.J. Leveque, "Stable Boundary Conditions for Cartesian Grid Calculations". NASA CR-182048, 1990.

[7] N. Bettschart, "Rotor Fuselage Interaction: Euler and Navier-Stokes Computations with an Actuator Disk". in *55th Annual Forum of the American Helicopter Society*. Montreal, Canada, 1999.

[8] P. Bradshaw, "Compressible Turbulent Shear Layers". *Annual Review of Fluid Mechanics*, **9**: 33-54, 1977.

[9] A.G. Brand, *An Experimental Investigation of the Interaction Between a Model Rotor and Airframe in Forward Flight*. PhD Thesis, Georgia Institute of Technology, Atlanta, 1989.

[10] J.R. Carlson, "High Reynolds Number Analysis of Flat Plate and Separated Afterbody Flow Using Non-linear Turbulence Models". AIAA Paper 96-2544, 1996.

[11] F. Casalini and A. Dadone, "Computations of Viscous Flows Using a Multigrid Finite Volume Lamda Formulation". *Engineering Computations*, **16**(7): 767-786, 1999.

[12] M.S. Chaffin and J.D. Berry, "Helicopter Fuselage Aerodynamics Under a Rotor by Navier-Stokes Simulation". *Journal of American Helicopter Society*, **42**(3): 235-242, 1997.

[13] M.S. Chaffin and J.D. Berry, "Navier-Stokes Simulation of a Rotor Using a Distributed Pressure Disk Method". in *51st Annual Forum of the American Helicopter Society*. Fort Worth, TX, 1995.

[14] E.F. Charlton, *An Octree Solution to Conservation-laws over Arbitrary Regions (OSCAR) with Applications to Aircraft Aerodynamics*. PhD Thesis, University of Michigan, Ann Arbor, MI, 1997.

[15] T. Chitsomboon, "Effects of Artificial Viscosity on the Accuracy of High-Reynolds-Number k-ε Turbulence Model". NASA TM 106781, 1994.

[16] T. Chitsomboon, "Improved Artificial Viscosity for High-Reynolds-Number κ–ε Turbulence Model". AIAA Paper 95-2166, 1995.

[17] A. Dadone and B. Grossman, "Far-Field Coarsening and Mesh Adaptation for Inviscid Flows on Cartesian Grids". AIAA Paper 2004-0585, 2004.

[18] A. Dadone and B. Grossman, "Further Developments in the Three-Dimensional Cartesian-Grid Ghost-Cell Method ". AIAA Paper 2006-1085, 2006.

[19] A. Dadone and B. Grossman, "Ghost-Cell Method for Inviscid Three-Dimensional Flows on Cartesian Grids ". AIAA Paper 2005-874, 2005.

[20] A. Dadone and B. Grossman, "An immersed body methodology for inviscid flows on Cartesian grids ". AIAA Paper 2002-1059, 2002.

[21] A. Dadone and B. Grossman, "Rapid Aerodynamic Optimization Using Far-Field Coarsened Cartesian Grids ". AIAA Paper 2004-4477, 2004.

[22] W. Dietz, et al., "Application of Vorticity Confinement to the Prediction of Flows over Complex Bodies". AIAA Paper 2001-2642, 2001.

[23] M. Fan and J. Steinhoff, "Computation of Blunt Body Wake Flow By Vorticity Confinement". AIAA Paper 2004-592, 2004.

[24] I. Fejtek and L. Roberts, "Navier-Stokes Computation of Wing/Rotor Interaction for a Tilt Rotor in Hover". *AIAA Journal*, **30**(11): 2595-2603, 1992.

[25] V.G. Ferreira, et al., "Numerical Simulation of Turbulent Free Surface Flow with Two-Equation κ–ε Eddy-Viscosity Models". *International Journal for Numerical Methods in Fluids*, **44**: 347-375, 2004.

[26] J.H. Ferziger, "Simulation of Incompressible Turbulent Flows". *Journal of Computational Physics*, **69**: 1-48, 1987.

[27] H. Forrer and R. Jeltsch, "A Higher-Order Boundary Treatment for Cartesian-Grid Method". *Journal of Computational Physics*, **140**: 259-277, 1998.

[28] C.E. Freeman and R.E. Mineck, "Fuselage Surface Pressure Measurements of a Helicopter Wind-Tunnel Model with a 3.15-Meter Diameter Single Rotor". NASA TM 80051, 1979.

[29] N.T. Frink, "Recent Progress Toward a Three-Dimensional Navier-Stokes Flow Solver". AIAA Paper 94-0061, 1994.

[30] G.A. Gerolymos, "Implicit Multiple-Grid Solution of the Compressible Navier-Stokes Equations Using k-ε Turbulence Closure". *AIAA Journal*, **28**(10): 1707-1717, 1990.

[31] A. Gessow and R.J. Tapscott, "Charts for Estimating Performance of High-Performance Helicopters". NACA Report 1266, 1956.

[32] D. Goldstein, R. Handler, and L. Sirovich, "Direct Numerical Simulation of Turbulent Flow over a Modelled Riblet Covered Surface". *Journal of Computational Physics*, **302**(354), 1995.

[33] J.J. Gorski, "A New Near-Wall Formulation for the κ–ε Equations of Turbulence". AIAA Paper 86-0556, 1986.

[34] P.A. Govatsos and D.E. Papantonis, "A Characteristic Based Method for the Calculation of Three-dimensional Incompressible, Turbulent and Steady Flows in Hydraulic Turbomachines and Installations". *International Journal for Numerical Methods in Fluids*, **34**: 1-30, 2000.

[35] J. Gullbrand, X.Z. Bai, and L. Fuchs, "Large Eddy Simulation of Turbulent Reacting Flows Using Cartesian Grid and Boundary Corrections". AIAA Paper 98-3317, 1998.

[36] N. Hariharan and L. Sankar, "Numerical Simulation of the Fuselage-Rotor Interaction Phenomenon". AIAA Paper 96-0672, 1996.

[37] D.G. Holmes and S.D. Connell, "Solution of the 2D Navier-Stokes Equations on Unstructured Adaptive Grids". AIAA Paper 89-1932-CP, 1989.

[38] D.L. Hunt and N.E. May, "Practical Use of Transport Turbulence Models in Aerospace - CFD Implementation and Applications -". AIAA Paper 99-3137, 1999.

[39] F. Ilinca and D. Pelletier, "Positivity Preservation and Adaptive Solution for the κ–ε Model of Turbulence". *AIAA Journal*, **36**(1): 44-50, 1998.

[40] F. Jacon and D. Knight, "Navier-Stokes Algorithm for Turbulent Flows Using an Unstructured Grid and Flux Difference Splitting". AIAA Paper 94-2292, 1994.

[41] I.K. Jennions and M.G. Turner, "Three-Dimensional Navier-Stokes Computations of Transonic Fan Flow Using an Explicit Flow Solver and an Implicit κ–ε Solver". ASME Paper 92-GT-309, 1992.

[42] W.G. Joo and T.P. Hynes, "The Application of Actuator disk to Calculations of the Flow in Turbofan Installations". *ASME Journal of Turbomachinery*, **119**(4), 1997.

[43] W.G. Joo and T.P. Hynes, "The Simulation of Turbomachinery Blade Rows in Asymmetric Flow Using Actuator Disks". *ASME Journal of Turbomachinery*, **119**(4), 1997.

[44] C.S. Kim, C. Kim, and O. H. Rho, "Parallel Computations of High-Lift Airfoil Flows Using Two-Equation Turbulence Models". *AIAA Journal*, **38**(8): 1360-1368, 2000.

[45] M.P. Kirkpatrick, S.W. Armfield, and J.H. Kent, "A Representation of Curved Boundaries for the Solution of the Navier-Stokes Equations on a Staggered Three-Dimensional Cartesian Grid". *Journal of Computational Physics*, **184**: 1-36, 2003.

[46] R.F. Kunz and B. Lakshminarayana, "Stability of Explicit Navier-Stokes Procedures Using k-ε and k-ε/Algebraic Reynolds Stress Turbulence Models". *Journal of Computational Physics*, **103**: 141-159, 1992.

[47] O.J. Kwon and C. Hah, "Simulation of Three-Dimensional Turbulent Flows on Unstructured Meshes". *AIAA Journal*, **33**(6): 1081-1089, 1995.

[48] M.C. Lai and C.S. Peskin, "An Immersed Boundary Method with Formal Second-order Accuracy and Reduced Numerical Viscosity". *Journal of Computational Physics*, **160**(2): 705-719, 2000.

[49] B.E. Launder and D.B. Spalding, "The Numerical Computation of Turbulent Flows". *Computer Methods in Applied Mechanics and Engineering*, **3**: 269-289, 1974.

[50] J. Lee and O.J. Kwon, "Predicting Aerodynamic Rotor-Fuselage Interactions by Using Unstructured Meshes". *Transactions of The Japan Society for Aeronautical and Space Sciences*, **44**(146): 208-216, 2002.

[51] B. van Leer, "Towards the Ultimate Conservative Difference Scheme III: Upstream-Centered Finite Difference Schemes for Ideal Compressible Flow". *Journal of Computational Physics*, **23**: 263-275, 1977.

[52] H. Luo, J.D. Bau, and R. Loehner, "Computation of compressible flows using a two-equation turbulence model of unstructured grids". AIAA Paper 97-0430, 1997.

[53] D. Marshall and S. Ruffin, "An Embedded Boundary Cartesian Grid Scheme for Viscous Flows Using a New Viscous Wall Boundary Condition Treatment ". AIAA Paper 2004-581, 2004.

[54] D. Marshall and S. Ruffin, "A New Inviscid Wall Boundary Condition Treatment for Embedded Boundary Cartesian Grid Schemes ". AIAA Paper 2004-583, 2004.

[55] D.D. Marshall, *Extending the Functionalities of Cartesian Grid Solvers: Viscous Effects Modeling and MPI Parallelization*. PhD Thesis, Georgia Institute of Technology, Atlanta, GA, 2002.

[56] D.J. Mavriplis and L. Martinelli, "Multigrid Solution of Compressible Turbulent Flow on Unstructured Meshes Using a Two-Equation Model". AIAA Paper 91-0237, 1991.

[57] D.M. McQueen and C.S. Peskin, "A Three-Dimensional Computational Method for Blood Flow in the Heart. II. Contractile Fibers". *Journal of Computational Physics*, **82**(2): 289-297, 1989.

[58] R.E. Mineck and S.A. Gorton, "Steady and Periodic Pressure Measurements on a Generic Helicopter Fuselage Model in the Presence of a Rotor". NASA TM-2000-210286, 2000.

[59] B. Mohammadi and O. Pironneau, "Unsteady Separated Turbulent Flows Computation with Wall-Laws and k-ε Model". *Computer Methods in Applied Mechanics and Engineering*, **148**: 393-405, 1997.

[60] J. Morrison, "Flux Difference Split Scheme for Turbulent Transport Equations". AIAA Paper 90-5251, 1990.

[61] S.M. Murman, M.J. Aftosmis, and M.J. Berger, "Implicit Approaches for Moving Boundaries in a 3-D Cartesian Method". AIAA Paper 2003-1119, 2003.

[62] R.H. Nichols and C.C. Nelson, "Wall Function Boundary Conditions Including Heat Transfer and Compressibility". *AIAA Journal*, **42**(6): 1107-1114, 2004.

[63] D.M. O'Brien and M.J. Smith, "Analysis of Rotor-Fuselage Interactions Using Various Rotor Models". AIAA Paper 2005-468, 2005.

[64] D.M. O'Brien and M.J. Smith, "Improvements in the Computational Modeling of Rotor/Fuselage Interactions Using Unstructured Grids". in *61st Annual Forum of the American Helicopter Society*. Grapevine, TX, 2005.

[65] D.M. O'Brien and M.J. Smith, "Understanding the Physical Implications of Approximate Rotor Methods Using an Unstructured CFD Method". in *31st European Rotorcraft Forum*, 2005.

[66] Y.M. Park, H.J. Nam, and O.J. Kwon, "Simulation of Unsteady Rotor-Fuselage Interaction Using Unstructured Adaptive Meshes". in *59th Annual Forum of the American Helicopter Society*. Phoenix, AZ, 2003.

[67] C.S. Peskin and D.M. McQueen, "A Three-Dimensional Computational Method for Blood Flow in the Heart. I. Immersed Elastic Fibers in a Viscous Incompressible Fluid". *Journal of Computational Physics*, **81**(2): 372-405, 1989.

[68] S.B. Pope, *Turbulent Flows*. Cambridge University Press, 2000.

[69] R.G. Rajagopalan and S.R. Mathur, "Three-Dimensional Analysis of a Rotor in Forward Flight". *Journal of American Helicopter Society*, **38**: 14-25, 1993.

[70] B.R. Ramaprian, V.C. Patel, and D.H. Choi, "Mean-Flow Measurements in the Three-Dimensional Boundary Layer over a Body of Revolution at Incidence". *Journal of Fluid Mechanics*, **103**: 479-504, 1981.

[71] T. Renaud, et al., "Evaluation of Isolated Fuselage and Rotor-Fuselage Interaction Using CFD". in *60th Annual Forum of the American Helicopter Society*. Baltimore, MD, 2004.

[72] A.M. Roma, C.S. Peskin, and M.J. Berger, "An Adaptive Version of the Immersed Boundary Method". *Journal of Computational Physics*, **153**(2): 509-534, 1999.

[73] E.M. Saiki and S. Biringen, "Numerical Simulation of a Cylinder in Uniform Flow: Application of a Virtual Boundary Method". *Journal of Computational Physics*, **123**: 450-465, 1996.

[74] T. Shih, K. Chen, and N. Liu, "A Non-linear κ–ε Model for Turbulent Shear Flows". AIAA Paper 98-3983, 1998.

[75] T.H. Shih, et al., "A Vorticity Dynamics Based Model for the Turbulent Dissipation". AIAA Paper 94-0194, 1994.

[76] D. L. Sondak and R.H. Pletcher, "Application of Wall Functions to Generalized Nonorthogonal Curvilinear Coordinate System". *AIAA Journal*, **33**(1), 1995.

[77] D.B. Spalding, "A Single Formula for the Law of the Wall". *Journal of Applied Mechanics*, **28**(3): 444-458, 1961.

[78] J. Steinhoff and M. Fan, "Vorticity Confinement for Computing Small Scales in Complex Flows, Including LES". AIAA Paper 2003-4099, 2003.

[79] J. Steinhoff and D. Underhill, "Modification of the Euler Equations for Vorticity Confinement: Application to the Computation of Interacting Vortex Rings". *Physics of Fluid*, **6**(8): 2738-2744, 1994.

[80] J. Steinhoff, Y. Wenren, and L. Wang, "Efficient Computation of Separating High Reynolds Number Incompressible Flows Using Vorticity Confinement". AIAA Paper 1999-3316, 1999.

[81] J.C. Tannehill, D.A. Anderson, and R.H. Pletcher, *Computational Fluid Mechanics and Heat Transfer*. 2nd Ed., Taylor & Francis, 1997.

[82] H. Tennekes and J.L. Lumley, *A First Course in Turbulence*. MIT Press, 1974.

[83] Y. Tseng and J.H. Ferziger, "A Ghost-Cell Immersed Boundary Method for Flow in Complex Geometry". *Journal of Computational Physics*, **192**: 593-623, 2003.

[84] S. Tu, *Development of a Solution Adaptive Cartesian-Grid Solver for 2-D Thermochemical Nonequilibrium Flows*. PhD Thesis, Georgia Institute of Technology, Atlanta, GA, 2001.

[85] H.S. Udaykumar, et al., "Multiphase Dynamics in Arbitrary Geometries on Fixed Cartesian Grids". *Journal of Computational Physics*, **137**(2): 366-405, 1997.

[86] H.S. Udaykumar, et al., "A Sharp Interface Cartesian Grid Method for Simulating Flows with Complex Moving Boundaries". *Journal of Computational Physics*, **174**: 345-380, 2001.

[87] R. Verzicco, et al., "Large Eddy Simulation in Complex Geometry Configurations Using Boundary Body Forces". *AIAA Journal*, **38**: 427-433, 2000.

[88] J.R. Viegas, M.W. Rubesin, and C.C. Horstman, "On the Use of Wall Functions as Boundary Conditions for Two-Dimensional Separated Compressible Flow". AIAA Paper 1985-180, 1985.

[89] Z.J. Wang and R.F. Chen, "Anisotropic Solution-Adaptive Viscous Cartesian Grid Method for Turbulent Flow Simulation ". *AIAA Journal*, **40**(10): 1969-1978, 2002.

[90] Z.J. Wang and X. Chi, "Direct Simulation of Surface Roughness Effects with a RANS and DES Approach on Viscous Adaptive Cartesian Grids ". AIAA Paper 2004-2420, 2004.

[91] J.M. Weiss and W.A. Smith, "Preconditioning Applied to Variable and Constant Density Flows". *AIAA Journal*, **33**(11): 2050-2057, 1995.

[92] Y. Wenren, et al., "Efficient Eulerian computation of realistic rotorcraft flows using Vorticity Confinement - A survey of recent results ". AIAA Paper 2001-996, 2001.

[93] Y. Wenren and J. Steinhoff, "Application of Vorticity Confinement to the Prediction of the Wake of Helicopter Rotors and Complex Bodies". AIAA Paper 1999-3200, 1999.

[94] Y. Wenren, et al., "Application of Vorticity Confinement to the Prediction of Flow over Complex Bodies". AIAA Paper 2000-2621, 2000.

[95] F.M. White, *Viscous Fluid Flow*. 2nd Ed., McGraw-Hill, 1991.

[96] D.L. Whitfield and A. Jameson, "Three Dimensional Euler Euqation Simulation of Propeller-Wing Interaction in Transonic Flow". AIAA Paper 83-0236, 1983.

[97] K. Wieghardt and W. Tillmann, "On the Turbulent Friction Layer for Rising Pressure". NACA TM 1314, 1951.

[98] T. Ye, et al., "An Accurate Cartesian Grid Method for Viscous Incompressible Flows with Complex Immersed Boundaries". *Journal of Computational Physics*, **156**: 209-240, 1999.

[99] D. Zeeuw, *A Quadtree-Based Adaptively-Refined Cartesian-Grid Algorithm for Solution of the Euler Equations*. PhD thesis, The University of Michigan, 1993.

[100] Y. Zhao, "Stable Computation of Turbulent Flows with a Low-Reynolds-Number k-ε Turbulence model and Explicit Solver". *Advances in Engineering Software*, **28**: 487-499, 1997.

[101] L.D. Zhu and C.S. Peskin, "Simulation of a Flapping Flexible Filament in a Flowing Soap Film by the Immersed Boundary Method". *Journal of Computational Physics*, **179**(2): 452-468, 2002.

[102] L.A. Zori, S.R. Mathur, and R.G. Rajagopalan, "Three-Dimensional Calculations of Rotor-Airframe Interaction in Forward Flight". in *48th Annual Forum of the American Helicopter Society*. Washington, D.C., 1992.

[103] L.A. Zori and R.G. Rajagopalan, "Navier-Stokes Calculations of Rotor-Airframe Interaction in Forward Flight". *Journal of American Helicopter Society*, **40**: 57-67, 1995.

VITA

Jae-Doo Lee was born in Seoul, Korea on February 1, 1973. He received his B.S. degree in Mechanical Engineering from Yonsei University, Seoul, Korea in February 1998. He then entered Yonsei Graduate School of Engineering. During the M.S. degree course, his graduate work is on the Computational Fluid Dynamics for the analysis of a compressor flow field. After receiving his M.S. degree in Mechanical Engineering in February 2000, he worked at Korea Institute of Science and Technology, Seoul, Korea. With an interest in an external compressible flow, he entered the doctoral program in the School of Aerospace Engineering at Georgia Institute of Technology in Atlanta, Georgia, USA in August 2001.